APPLYING A NODAL DIFFUSION-MICRO DEPLETION SEQUENCE FOR HIGH-FIDELITY ANALYSIS OF SODIUM COOLED FAST REACTORS

A Thesis
Presented to
The Academic Faculty

by

Gustavo Sigwalt Horn Pereira

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science in
Nuclear and Radiological Engineering

Georgia Institute of Technology
December 2018

APPLYING A NODAL DIFFUSION-MICRO DEPLETION SEQUENCE FOR HIGH-FIDELITY ANALYSIS OF SODIUM COOLED FAST REACTORS

Approved by:

Dr. Dan Kotlyar, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Bojan Petrovic
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Nolan E. Hertel
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: November 20, 2018

I dedicate this thesis to all of my family, friends, colleagues and professors, all of whom played a key and unique role in my journey.

ACKNOWLEDGEMENTS

I would first and foremost like to acknowledge my thesis advisor, Dr. Dan Kotlyar, whose willingness and availability to help me made the success of this thesis possible. Dr. Kotlyar gracefully took me in his group and gave me all the support and resources I needed to complete this work. I would also like to thank Dr. Bojan Petrovic and Dr. Nolan Hertel for participating in my thesis committee and providing valuable feedback on my work.

As I've now learned over my years in college and in graduate school, a key component to success involves the people that surround you on a daily basis. And for that, I have to acknowledge my classmates and lab mates, in particular Andrew Johnson, Jim Wang, Stefano Terlizzi, Vedant Mehta and Will DeShazer. Whether celebrating the end of a semester over a cold beverage, struggling with a homework problem deep into the night, or even just making a joke during the day to relieve some of the stress related to being a grad student, those guys were always by my side and willing to help whenever I needed it.

One of the reasons I chose Georgia Tech for my graduate studies was to be closer to my family. Dad, mom, brother, sister, you have always provided me with the support I needed, and being close to you made this journey much smoother than it otherwise would have been had I gone far away. I am forever grateful for the love and opportunities you all have given me.

Last, but certainly not least, I need to thank the support from my friends Chris and Aram, as well as the people who kept me company during the numerous hours I spent in

Atlanta traffic, especially Bill, Joe, Theo, Joey, Bert, Jordan, Ari, Tom and the main fgtrtd Christina. You guys always encouraged me to try it out and go full throttle while progressing through this journey, and although most problems took much more than four strokes to solve, I have finally sharpened my tools well enough to feel confident as I face new challenges moving forward. I like that.

The work presented here was partially funded through the Nuclear Regulatory Commission project number NRC-HQ-84-14-G-0058.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
ABR	Advanced Burner Reactor
B1	Leakage Corrected Cross Sections
BoC	Beginning of Cycle
DH	Decay Heat
EoC	End of Cycle
EoS	End of Step
GNEP	Global Nuclear Energy Parternship
INF	Infinite Spectrum Cross Sections
LWR	Light Water Reactor
MC	Monte Carlo
MIT	Massachusetts Institute of Technology
MWth	Mega-Watt Thermal
NEA	Nuclear Energy Agency
NEM	Nodal Expansion Method
OECD	Organisation for Economic Co-operation and Development
OpenMP	Open Multi-Processing
PWR	Pressurized Water Reactor
RMS	Root Mean Square
SFR	Sodium Fast Reactor
SPH	Superhomogenization
T/H	Thermal Hydraulic
TRU	Transuranic

SUMMARY

Recent developments in the global nuclear industry have led to the need of reactor designs that are not only safe, but also address the challenges involving nuclear waste while producing clean electricity at low costs. One of the designs proposed to fill these requirements is the Advanced Burner Reactor (ABR), a sodium cooled, metal fuel fast reactor system that uses spent fuel from current light water reactors as part of its energy source. Due to the complex nature of nuclear reactors, extensive modeling of a system must be performed in order to demonstrate the viability of such system.

This thesis combines two established reactor modeling techniques in order to efficiently model the ABR core. The computational methods used in this work are Monte Carlo (MC) and nodal diffusion. The MC method is a well-established computational approach for modeling of nuclear systems, and is considered to be very accurate and versatile. However, the MC requires extensive time and computational resources, and its applicability becomes prohibitively expensive when performing analyses of accident scenarios. Meanwhile, the nodal diffusion method requires much fewer resources to perform such analyses, but theoretically the accuracy is compromised due to the simplifications applied to the model.

The main focus of the work presented in this thesis revolves around expanding the capabilities of nodal diffusion codes to calculate local isotopic concentrations, activities and decay heat quantities, which is a first-of-a-kind demonstration of the applicability of nodal diffusion codes for such calculations. Establishing this approach allows for the

capability of decay heat to be calculated rapidly and efficiently, allowing for the performance of transient analyses in accident scenarios.

The work presented in this thesis uses the MC code Serpent as a macroscopic and microscopic cross-section generation tool, and the nodal diffusion code DYN3D for full core analysis of the ABR core. The Serpent-DYN3D code sequence is then applied for various scenarios, including decay heat analysis, and compared to reference MC solutions. It is found that the Serpent-DYN3D sequence is an adequate tool for modeling of sodium cooled, metal fuel fast reactors, providing accurate solutions while saving on time and computational resources required.

CHAPTER 1. INTRODUCTION

This thesis investigates an extended coupling of the Monte Carlo and nodal diffusion computational methods for reactor physics analysis of metal fuel, sodium cooled fast reactors. The coupling is extended by exploring the capability of nodal diffusion codes to calculate local isotopic concentrations, activities and decay heat quantities. This document presents the methodology used to perform various analyses related to core performance, including static state, depletion and decay heat analysis. The following sections will lay out the motivation for the work, the objectives of the thesis, and a description of the work performed.

1.1 Motivation

One of the major hurdles faced by the nuclear industry is the need to find a solution for the spent nuclear fuel generated in light water reactors (LWRs). However, the fuel discharged from LWRs still contains a large amount of energy, and can be used as fuel in alternative reactor designs, such as fast reactors. The sodium cooled fast reactor (SFR) is one of the designs being developed as part of the Generation IV reactors, which aims at increasing performance, safety and economic viability of nuclear technology [1]. Due to the complex nature of nuclear reactors, detailed modeling is necessary to validate reactor designs before any experiments or construction can take place. Throughout the last half century, computational methods have continuously been developed to be able to model even the most complex reactor performance. The Monte Carlo method is a well-established method that is known to provide accurate results for neutronics analysis. However, this method is very costly, both computationally and time wise. The computational

requirements are further amplified for complex and large geometries, such as in the case of full core analysis. An alternative approach to the Monte Carlo method relies on the reduced-order nodal diffusion methodology. Nodal diffusion codes are generally less accurate than Monte Carlo codes, but are able to produce results much faster and with much lower computational costs. In addition, nodal codes have no knowledge of micro quantities, such as concentrations and spatial decay heat distributions. The traditional lattice-nodal diffusion sequence relies on generating *macroscopic* cross-sections, and limited number of microscopic cross-sections (*e.g.*, xenon and samarium), and feeding these into nodal solvers. A primary focus of this thesis is to extend this approach by including a full set of micro cross-sections for multiple isotopes (~290) that enable to track the spatial decay heat production within the core. Motivated by developing a scheme that provides accurate results while requiring fewer computational resources and shorter time, the applicability of coupling Monte Carlo and nodal diffusion methods for the analysis of a metal fuel, sodium cooled fast reactor is investigated in this thesis.

1.2 Objectives

The main objective of the thesis is to show that the metal fuel, sodium cooled fast reactor can be modeled accurately by the Monte Carlo-nodal diffusion sequence, and yet require considerably less computational resources when compared to full core analysis with solely Monte Carlo methods. This is accomplished by:

- Developing a Serpent model of the ABR core to be used as a reference solution.
- Establishing the Serpent-DYN3D code sequence for basic depletion problems.
- Generating macroscopic and microscopic cross section libraries using Serpent.

- Applying the Seprent-DYN3D for spatial decay heat generation for varying levels of geometric complexity, including full core 3D models.

In order to validate the approach, the results obtained are compared to MC Serpent models, which are taken as reference solutions.

1.3 Scope

As a starting point, the first portion of the work relates to the OECD "Benchmark for Neutronic Analysis of Sodium-cooled Fast Reactor Cores with Various Fuel Types and Core Sizes" [2]. The medium size, 1000 MW thermal, metal fuel Advanced Burner Reactor (ABR) core is chosen as the reference design for this thesis. All Monte Carlo simulations presented in this work were performed using Serpent.

First, a 3D model of the full core was developed using Serpent, and results are compared to reported results from the benchmark. Once good agreement was achieved, the Serpent model was taken as the reference solution for comparison with nodal diffusion results. The nodal diffusion code used in the analysis is DYN3D. The Serpent-DYN3D code sequence is used to model the reactor by using Serpent as a cross section generator for small portions of the reactor, and DYN3D for full core analysis. Serpent-DYN3D code sequence is then used for static state analysis, depletion analysis, and decay heat analysis with micro depletion.

1.4 Thesis Layout

In Chapter 2, background information pertaining to the remaining of the thesis will be presented, including descriptions of the reactor physics concepts, computational

methods and computer codes used. Chapter 3 provides a detailed description of the ABR core geometry and materials used. In Chapter 4, the Serpent model used to validate Serpent versus the results reported in the benchmark is described and results are presented. Chapter 5 establishes the Serpent-DYN3D code sequence for 2D and 3D simple depletion problems, and the methodology used for macroscopic cross-section generation is described. Chapter 6 contains the bulk of the work, where first the microscopic cross-section generating process is presented, followed by an analysis of operational conditions effects on decay heat generated, and finally a full decay heat analysis for multiple geometries is performed applying the Serpent-DYN3D code sequence. Finally, Chapter 7 presents a set of conclusions and some possible future work related to this thesis.

CHAPTER 2. BACKGROUND

In this chapter, an overview of the computational methods, computer codes, and the ABR reactor design will be presented. Information from this chapter will be used throughout the thesis.

2.1 Neutron Transport

A key aspect in modeling nuclear reactor cores is the understating of the behavior of the particles within the core, especially the distribution of neutrons in the reactor. The movement and interactions of neutrons is described by neutron transport theory. By understating the core configuration and material composition at an initial point in time, the behavior of the reactor can be predicted based on operating conditions and interactions between particles, and material compositions can be predicted at future points in time.

The mathematical model that describes the behavior of neutrons in the core is known as the neutron transport equation. This is a linear equation in the unknown dependent variable $n(\boldsymbol{r}, E, \widehat{\Omega}, t)$, known as angular neutron density, and seven independent variables: three in space $\boldsymbol{r} = (x, y, z)$, three in velocity (one in energy E, two in angular direction $\widehat{\Omega} = (\theta, \phi)$), and one in time t. The equation describes neutron balance in a medium, namely, the fact that the number of neutrons lost (through leakage and collision out of the velocity of interest) must equal the number of neutrons gained (through independent source, streaming, and collision into the velocity of interest). The most rudimentary form of the neutron transport equation is then described by Equation 2.1 [3]:

$$\frac{\partial n}{\partial t} + v\hat{\Omega} \cdot \nabla n + v\Sigma_t\, n(\boldsymbol{r}, E, \hat{\Omega}, t)$$
$$= \int_{4\pi} d\hat{\Omega}' \int_0^{\infty} dE'\; v'\Sigma_s\big(E' \longrightarrow E, \hat{\Omega}' \longrightarrow \hat{\Omega}\big) n\big(\boldsymbol{r}, E', \hat{\Omega}', t\big) + \; s(\boldsymbol{r}, E, \hat{\Omega}, t) \tag{2.1}$$

The neutron transport equation is often simplified and written in terms of the angular flux (ψ), and when considering systems without external sources, the criticality can be expressed in terms of the eigenvalue k , so that the neutron transport equation takes the form of Equation 2.2 [3]:

$$\frac{1}{v}\frac{\partial \psi}{\partial t} + \hat{\Omega} \cdot \nabla \psi + \Sigma_t(\boldsymbol{r}, E)\; \psi(\boldsymbol{r}, E, \hat{\Omega}, t)$$
$$= \int_{4\pi} d\hat{\Omega}' \int_0^{\infty} dE'\; \Sigma_s\big(E' \longrightarrow E, \hat{\Omega}' \longrightarrow \hat{\Omega}\big) \psi\big(\boldsymbol{r}, E', \hat{\Omega}', t\big) \tag{2.2}$$
$$+ \frac{X(E)}{k} \int_{4\pi} d\hat{\Omega}' \int_0^{\infty} dE'\; \Sigma_f(E') \psi\big(\boldsymbol{r}, E', \hat{\Omega}', t\big)$$

While the neutron transport equation is precise in tracking the behavior of neutrons throughout the reactor and throughout time, it is very complicated to obtain an explicit solution, even with modern day high computational capabilities. Therefore, a number of simplifications and assumptions can be made in order to make the equation easier to solve. One of the most common simplifications is to approximate the streaming term via the neutron diffusion theory [4].

2.2 Neutron Diffusion

Neutron diffusion is derived from a similar approach as used to model the diffusion of particles in liquids and gaseous environments. In essence, it predicts that particles (or in

reactor case, neutrons) will diffuse from regions of high particle density to regions of low particle density. Diffusion theory is based on Fick's law, which relates the neutron current and neutron flux by the following equation [4]:

$$\mathbf{J}(\mathbf{r}) = -D(\mathbf{r})\nabla\phi(\mathbf{r}) = -\frac{1}{3\Sigma_{\mathrm{tr}}(\mathbf{r})}\nabla\phi(\mathbf{r}) \tag{2.3}$$

Equation 2.3 describes that the neutron current, **J,** in the direction **r** , is proportional to the gradient of the flux (ϕ) with the proportionality parameter known as the diffusion coefficient, $D(\mathbf{r})$, which has units of centimeters. Applying this concept to multiple directions and considering neutron balance in a fashion similar to that described in Section 2.1 leads to the neutron diffusion equation [4]:

$$\frac{1}{v}\frac{\partial\phi}{\partial t} - \nabla\cdot(D\nabla\phi) + \Sigma_a\phi = s \tag{2.4}$$

If the medium is homogeneous, the term $\nabla\cdot(D\nabla\phi)$ can be replace by $D\nabla^2\phi$, where ∇^2 is the Laplacian, and v refers to the neutron speed.

Although the neutron diffusion equation allows for a much quicker and simpler solution for neutron distribution than the transport equation, it has its limitations. Neutron diffusion cannot be directly applied in regions where the flux gradient is large, such as near system boundaries and near sources or sinks. The following sections will describe the computational methods applied in the work presented in this thesis, as well as a brief introduction of the computer codes used.

2.3 Computational Methods

2.3.1 Monte Carlo

The Monte Carlo method is capable of providing high accuracy solutions to reactor analysis problems. While many simplified approaches use variable discretization to solve the complex transport equation, causing the solution to lose resolution due to the approximations made, the Monte Carlo method maintains maximum solution accuracy by preserving the continuous nature of the variables. By maintaining the continuous nature of the energy, spatial, and angular variables, the Monte Carlo method can be applied to accurately model systems containing highly complex geometries. This is accomplished by sampling random particles (in the case of nuclear core analysis, neutrons) individually, and tracking their behaviour in the core based on their interactions with core materials and boundaries. Because the particle interaction is simulated using statistical physical models, there is a degree of statistic uncertainty associated with Monte Carlo calculations. This uncertainty can be reduced by increasing the number of particles tracked in the system. However, increasing the number of particles tracked will lead to an increase in computational resources required. For some cases, the additional computational resources and amount of time required can be prohibitive, and alternate solutions must be used. This is especially true when Monte Carlo codes are coupled with depletion and thermal-hydraulic feedback. Typically, such integration requires to iteratively update local properties, such as temperatures, nuclide concentrations, and power and results in high computational resource requirements.

2.3.2 Nodal Diffusion

According to the nodal methodology it is assumed that a reactor can be spatially subdivided into a set of nodes; these are parts of the fuel assemblies divided in axial layers. Each node is characterized by the diffusion equation (Eq. 2.4) with unique set of multi-group cross-sections. The three-dimensional neutronic model utilizes nodal expansion methods (NEM) to solve the multi-group neutron diffusion equation for Cartesian and Hexagonal geometry. The nodal diffusion method requires considerably less computational resources when compared to the Monte Carlo method. However, nodal diffusion requires that a set of cross sections, pre-generated for a given node, be provided. These cross sections are usually generated using codes that solve the transport equation (such is the case for codes that apply Monte Carlo). Further description of cross section generation will be provided in future chapters. Using the homogenised few group constants, the 3D neutron diffusion equation is solved and each node is represented by an average neutron flux. In addition, parameters such as criticality and spatial power can be generated throughout the reactor. One limitation that nodal diffusion faces is found when two adjacent nodes have highly different flux profiles, *e.g.*, a fuel assembly next to a reflector assembly. This problem can be remediated by applying correction techniques, where the flux is re-evaluated based on a correction factor relating the homogeneous to the heterogeneous flux in that node. Finally, it must be pointed out that the results obtained by any nodal diffusion code can be only as good as the produced cross-section that represent a unique node.

2.4 Codes

This thesis heavily uses the Monte Carlo based Serpent code, developed in Finland, to perform transport calculations, and the DYN3D code, developed in Germany, for the nodal diffusion analysis. An overview of these codes, along with the description of tools used for data processing, is presented here.

2.4.1 DYN3D

The nodal diffusion code used in this work is DYN3D, a 3-dimensional nodal reactor dynamic code developed in Germany by Helmholtz-Zentrum Dresden-Rossendorf. The code was mainly developed for the analysis of transient scenarios for LWRs with hexagonal or square fuel lattices, but its functionality has been expanded for steady-state and fuel cycle calculations, including decay heat calculations [5], as well as for triangular assembly geometries [6]. Additionally, DYN3D has been shown to be a feasible tool for modelling of Sodium Fast Reactors (SFRs) for reactors with oxide fuel [7]. Although DYN3D can be coupled with a number of system codes, as well as thermal hydraulic and fuel performance codes, in order to expand its applicability, the work presented here relies only on neutronic analysis and thus no coupling of DYN3D with any other codes was realized. In order to be executed, DYN3D needs to be provided with a pre-generated cross-section library. In this work, cross section generation is done using the Monte Carlo code Serpent 2.

2.4.2 Serpent

Serpent is a multipurpose 3D Monte Carlo particle transport code developed at the VTT Technical Research Centre of Finland [8]. Serpent allows for the modelling of complex 3D geometries and can be used in fuel cycle analysis given its burnup capabilities. Originally, Serpent was developed mainly as a tool for generation of homogenized few-group constants, both for zero-leakage infinite lattices (*INF*) and for systems with non-zero leakage currents, in which spectrum correction schemes (*B1*) are applied [9]. Serpent is continuously being update and is now very attractive for multi-physics calculations specifically for its efficient Woodcock Delta-tracking routine [10]. In this work, Serpent 2 is used with the ENDF/B-VII cross section library as a few-group constant generator for homogenized macro and microscopic cross-sections, as well as for reference solutions for the cases analysed.

2.4.3 Processing Tools

While basic Excel spread sheets and MATLAB scripts were sufficient tools for data processing of the early portion of the work presented herein, the decay heat analysis involved manipulation of massive files that made usage of such tools prohibitive. Therefore, the serpent-tools package was used. The serpent-tools python package is a collection of parsing tools and containers aimed to expedite analysis of Serpent outputs. Files that would make MATLAB extremely slow or unresponsive can be processed within fractions of seconds, with no loss of data. The data is stored in an object-oriented framework that mimics the physical nature of the quantities represented, e.g. Detector objects have tallies and grid structures, Depleted Material objects have names and

associated atomic density, toxicity, and burnup matrices. Many of the readers and containers have routines for expediting common analyses, with heavy emphasis on plotting. Plots of Cartesian and Hexagonal detector meshes, flux spectra, homogenized group constants, and depletion parameters are made accessible to the user, without requiring a high learning curve. The project is hosted on GitHub with a permissive MIT license and is undergoing constant development and improvement. A thorough overview of the supported file types with examples can be found through the repository [11].

CHAPTER 3. ADVANCED BURNER REACTOR

The work presented herein uses the Advanced Burner Reactor (ABR) design proposed by Argonne National Laboratory as its focused reactor core [12], although the methods and code sequences used in this thesis should be applicable to any type of reactor design. The ABR is a 1000 MWth sodium cooled fast reactor with a metallic core. The core design was developed under the Global Nuclear Energy Partnership (GNEP) program. The ABR design proposes a transuranic (TRU) conversion ratio of approximately 0.7 and has a one-year fuel cycle with assumed 90% capacity factor. The materials used in the ABR design are constrained to conventional or reasonably proven materials in order to be in accordance with current fast reactor experience and expertise.

3.1 ABR Core Description

The ABR metallic core is composed of 2 radially unique fuel regions, as seen in Figure 3.1:

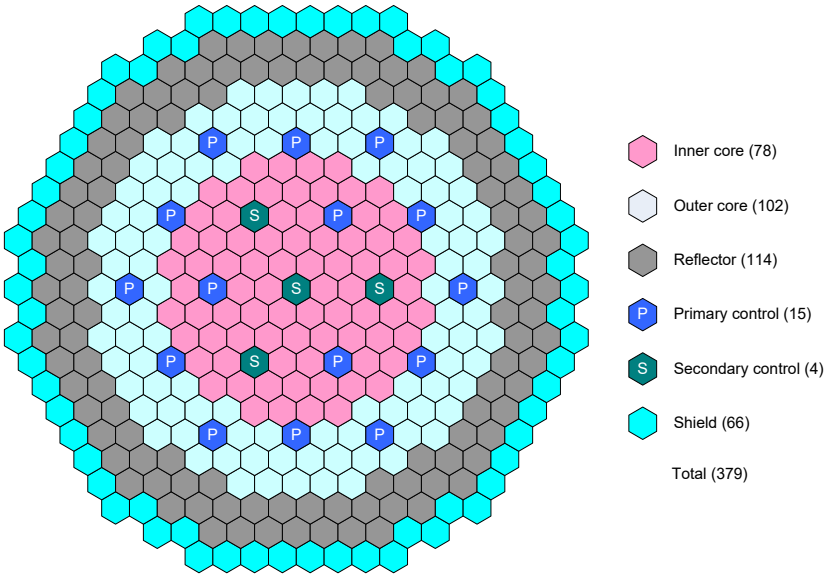


Figure 3.1 Radial Core Layout of ABR Metallic-Fuel Core [2]

The core is made up of 379 hexagonal assemblies, of which 180 are driver assemblies (78 inner core, 102 outer core), as well as 114 reflector, 19 control, and 66 shield subassemblies. Two independent safety control systems are implemented, with a primary control system containing 15 subassemblies, and a secondary control system containing 4 control subassemblies. Whereas, in reality, the core is surrounded by various materials, a vacuum boundary condition is adopted for outside of the shielding region in the benchmark and is replicated in all associated analyses presented in the thesis.

The nominal ABR power is 1000 MWth, with a coolant inlet and outlet temperatures of 355 °C and 510 °C, respectively. For simplicity, a constant average coolant, fuel and structural material temperature were used in the benchmark and have been used throughout the work performed herein. The used values are summarized in Table 3.1.

Table 3.1 Nominal Operating Condition, Metallic Core [2]

	Unit	Value
Reactor Power	MW-thermal	1000.0
Coolant temperature	°C	432.5
Average core structural temperature	°C	432.5
Average metallic fuel temperature	°C	534.0

3.1.1 Fuel Assembly

Detailed description of the driver assembly under nominal operating conditions is presented in Table 3.2:

Table 3.2 Parameters for Driver Subassembly of ABR Metallic Core [2]

	Unit	Operating state
Overall length of subassembly	cm	480.20
- Lower structure		35.76
- Lower reflector		125.16
- Active core height		85.82
- Replaced bond sodium		20.06
- Gas plenum		101.01
- Upper structure		112.39
Subassembly pitch, cm	cm	16.2471
Subassembly duct outer flat-to-flat distance	cm	15.8123
Subassembly duct wall thickness	cm	0.3966
Number of fuel pins		271
Outer radius of cladding	cm	[a] 0.3857
Inner radius of cladding	cm	0.3236
Fuel slug radius	cm	0.3236
Pin to Pin distance	cm	0.8966

a) Cladding outer radius has been slightly increased to compensate for the smearing of the wire wrap with the cladding.

The driver subassembly is composed of 271 cylindrical fuel pins, has an active fuel height of 85.82 cm, and is divided into 5 isometric axial regions of unique fuel composition. The gas plenum is included in order to accommodate gaseous fission products and allow the sodium to expand during operations, as the fuel slug grows 5% due to radiation swelling [2]. The axial schematics of the fuel subassembly included in the core is presented in the following figure:

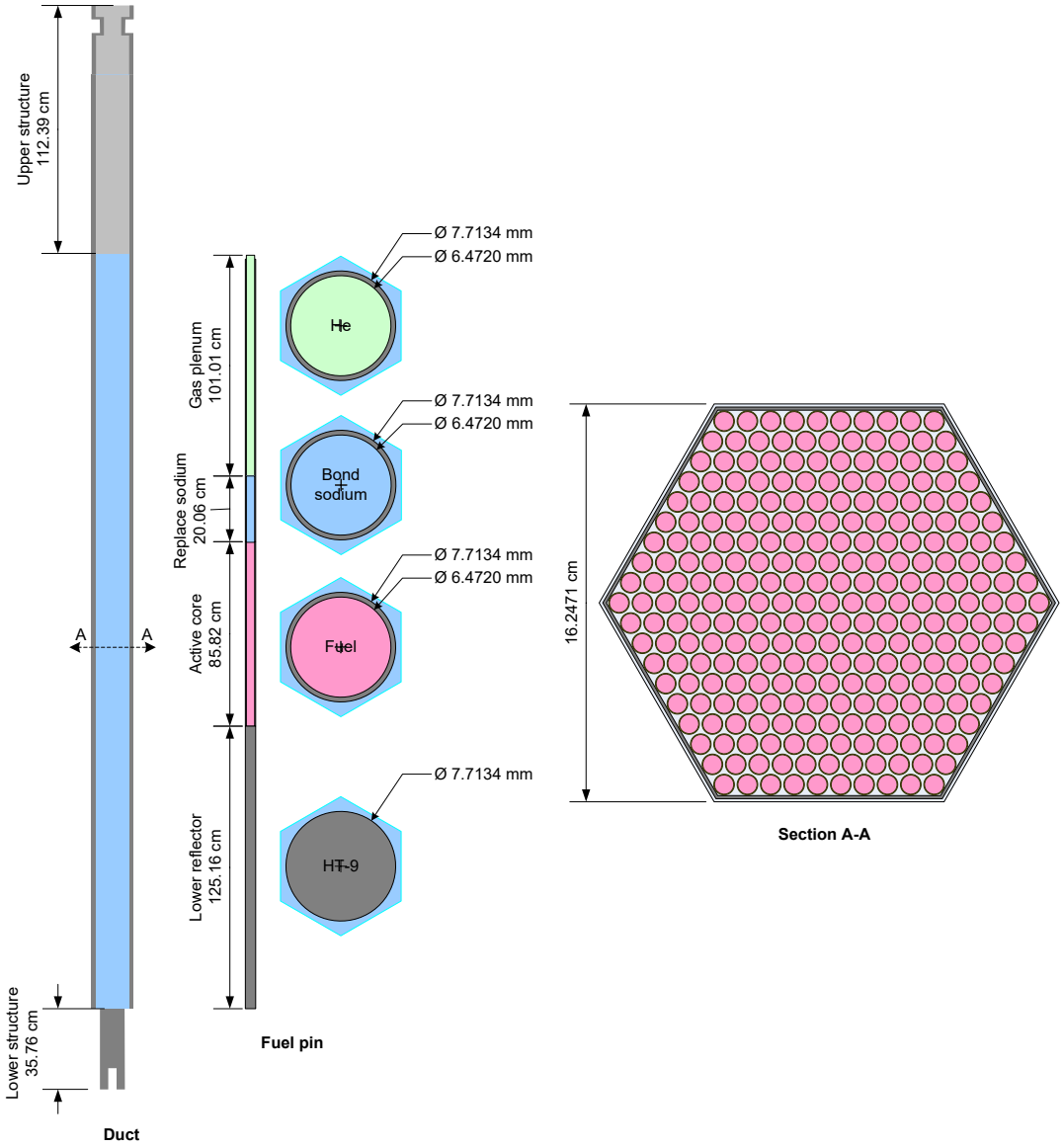


Figure 3.2 Schematics of Driver Subassembly of ABR Metallic Core [2]

3.1.2 Control Rod Assembly

Detailed description of the control rod subassembly dimensions is presented in Table 3.3:

Table 3.3 Parameters for Control Assembly of ABR Metallic Core [2]

	Unit	Operating state
Overall length of subassembly	cm	480.20
- Lower structure		35.76
- Lower reflector		125.16
- Absorber		86.75
- Empty duct		232.53
Subassembly pitch, cm	cm	16.2471
Subassembly duct outer flat-to-flat distance	cm	15.8123
Subassembly duct wall thickness	cm	0.3966
Interior duct outer flat-to-flat distance	cm	12.5125
Interior duct wall thickness	cm	0.3966
Number of fuel pins		19
Outer radius of cladding	cm	[a] 1.1693
Inner radius of cladding	cm	1.0977
Absorber radius	cm	1.0977
Pin to Pin distance	cm	2.458

a) Cladding outer radius is increased to compensate for the smearing of the wire wrap.

The control subassemblies are designed to control reactivity by introducing materials that are strong neutron absorbers into the core. These assemblies are divided into interior and outer ducts. The outer duct has the exact same dimensions as the fuel driver assembly, with the interior duct having smaller dimensions to allow for the free motion of the control rods. The interior duct contains 19 cylindrical control rods made of HT-9 tubes filled with boron carbide pellets. The differentiator between primary and secondary control assemblies is that the primary control rods use natural boron, while the secondary control rods use enriched boron (65% atomic fraction). The axial schematics of the control subassemblies of the ABR core is presented in Figure 3.3:

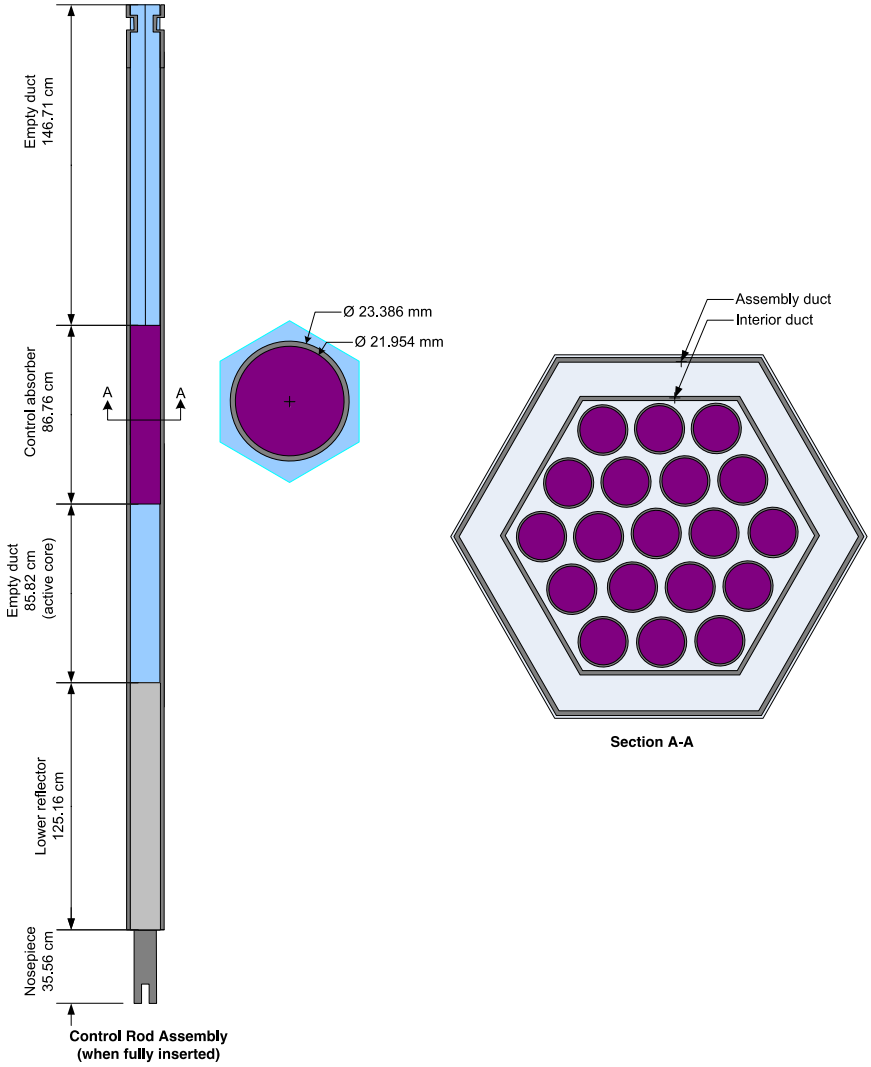


Figure 3.3 Schematics of Control Subassembly of ABR Metallic Core [2]

3.1.3 Radial Reflector

The radial reflector subassemblies are made up of 91 solid HT-9 pins, and serve the purpose of limiting neutron leakage at the boundary of the active core by reflecting neutrons back towards the inner portion of the reactor. The detailed description of the reflector subassemblies dimensions is provided in Table 3.4, with axial schematics presented in the following Figure 3.4:

Table 3.4 Parameters for Radial Reflector Subassembly of ABR Metallic Core [2]

	Unit	Operating state
Overall length of subassembly	cm	480.20
- Lower structure		35.76
- Lower reflector		125.16
- Radial reflector		206.89
- Upper structure		112.39
Subassembly pitch, cm	cm	16.2471
Subassembly duct outer flat-to-flat distance	cm	15.8123
Subassembly duct wall thickness	cm	0.3966
Number of fuel pins		91
Rod radius	cm	0.7757
Pin to Pin distance	cm	1.5528

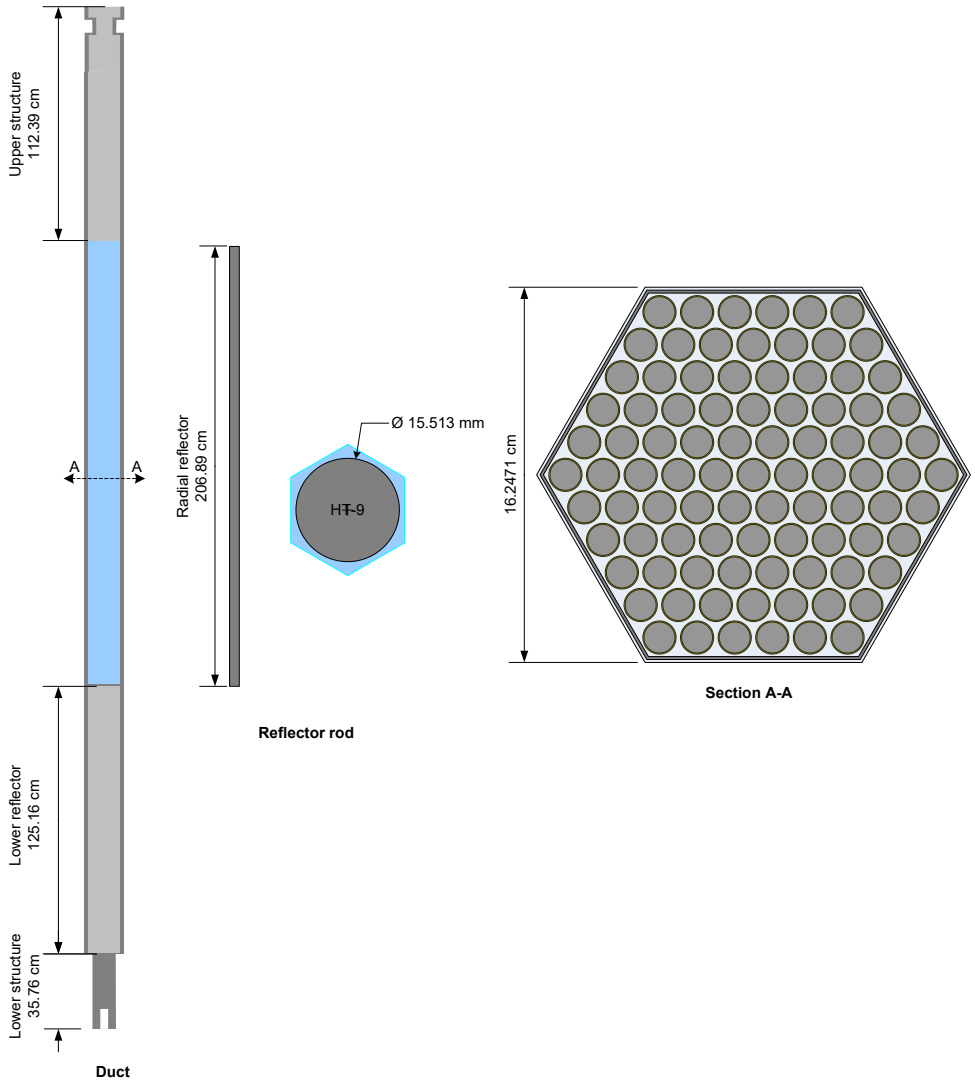


Figure 3.4 Schematics of Radial Reflector Subassembly of ABR Metallic Core [2]

3.1.4 Radial Shield

The ABR radial shield provides protection to the reactor's surroundings by using strong neutron absorbing materials. These assemblies consist of 19 thick HT-9 tubes, the same material used for cladding, each filled boron carbide pellets. Detailed description of the subassembly dimensions and axial schematics are presented in the following table and figure:

Table 3.5 Parameters for Shielding Subassembly of ABR Metallic Core [2]

	Unit	Operating state
Overall length of subassembly	cm	480.20
- Lower structure		35.76
- Lower reflector		125.16
- Radial shield		206.89
- Upper structure		112.39
Subassembly pitch, cm	cm	16.2471
Subassembly duct outer flat-to-flat distance	cm	15.8123
Subassembly duct wall thickness	cm	0.3966
Number of fuel pins		19
Outer radius of cladding	cm	1.6794
Inner radius of cladding	cm	1.4277
Absorber radius	cm	1.4277
Pin to Pin distance	cm	3.3603

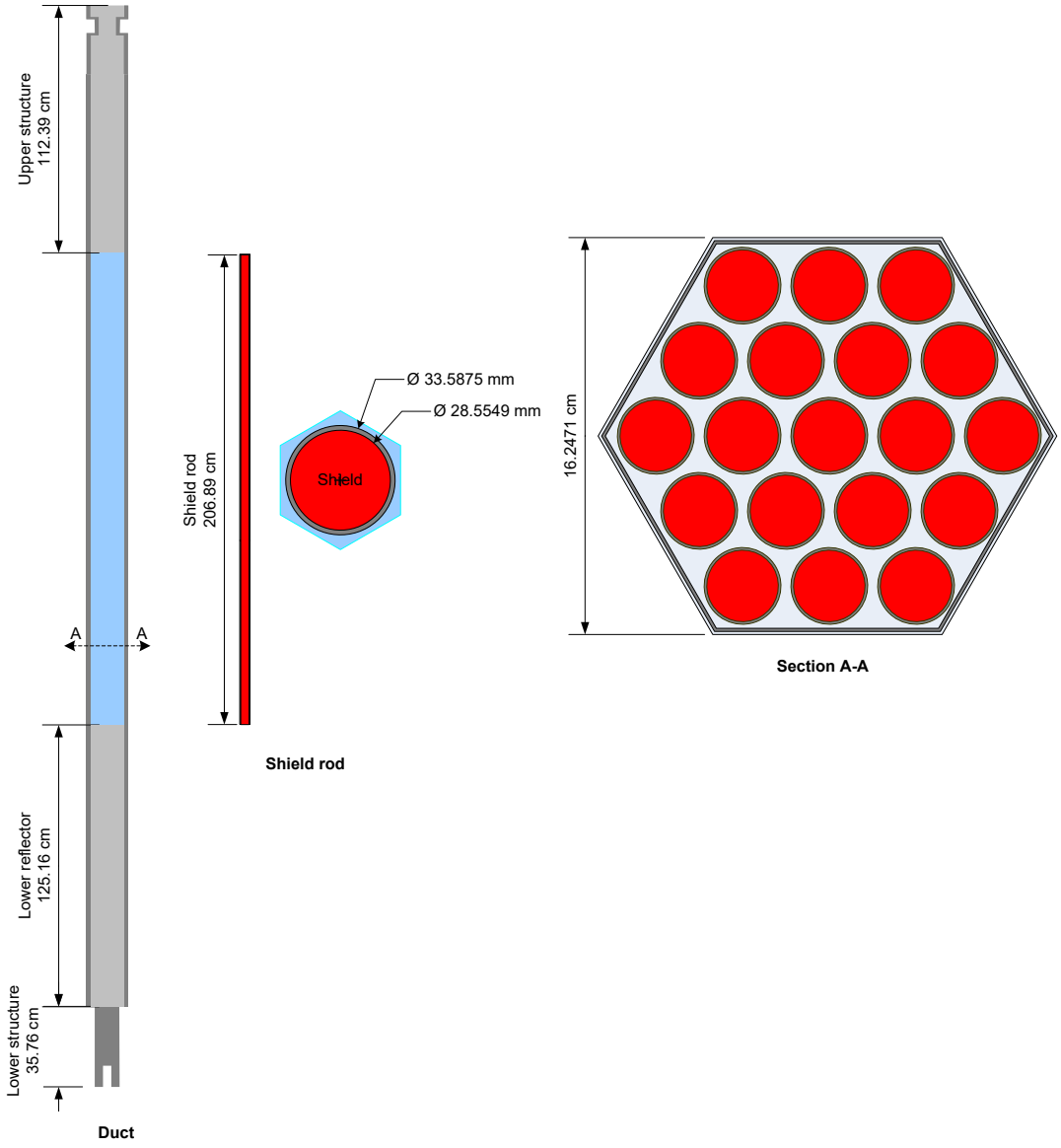


Figure 3.5 Schematics of Radial Shield Subassembly of ABR Metallic Core [2]

3.1.5 Full Core

An overview of the volume fraction that each type of material takes up in each of the subassemblies is presented in Table 3.6:

Table 3.6 Volume Fractions of ABR Metallic Core (%) [2]

Region		Coolant	HT-9	Fuel	Natural B_4C	Enriched B_4C
Lower structure		Homogeneous mixture of 30% SS-316 and 70% Sodium				
Lower reflector		35.34	64.66			
Upper structure		35.34	64.66			
Driver	Active core	35.34	25.66	39.00		
	Displaced bond sodium	74.34	25.66			
	Gas plenum	35.34	25.66			
Radial reflector		15.50	84.50			
Radial shield		17.10	29.68		53.23	
Absorber in primary control subassembly		47.75	20.78		31.46	
Absorber in secondary control subassembly		47.75	20.78			31.46
Empty duct in control subassembly		90.74	9.26			

3.2 ABR Material Description

The fuel composition is provided both at the beginning of cycle (BoC) and end of cycle (EoC) by the benchmark definition. The fuel is made up of recovered TRU material from discharged LWR fuel. The following tables provide the fuel composition for each axial layer both in the inner and outer core regions:

Table 3.7 Number Densities of Inner Core Fuel Pin, BOC (atoms/barn-cm)

Nuclide	Upper boundary from active core bottom (cm)				
	17.16	34.33	51.49	68.66	85.82
U-234	1.1369E-06	1.0856E-06	1.0727E-06	1.1028E-06	1.1759E-06
U-235	3.0421E-05	2.9338E-05	2.8961E-05	3.0070E-05	3.2571E-05
U-236	2.4896E-06	2.5117E-06	2.5536E-06	2.3779E-06	2.0226E-06
U-238	1.9613E-02	1.9474E-02	1.9433E-02	1.9550E-02	1.9801E-02
Np-237	4.6686E-05	4.6962E-05	4.6782E-05	4.7603E-05	4.8895E-05
Pu-236	4.9700E-10	5.5883E-10	5.6701E-10	5.5075E-10	4.8775E-10
Pu-238	1.1695E-04	1.1284E-04	1.1196E-04	1.1370E-04	1.1829E-04
Pu-239	2.2076E-03	2.1814E-03	2.1754E-03	2.1813E-03	2.2011E-03
Pu-240	1.3244E-03	1.2955E-03	1.2902E-03	1.2986E-03	1.3248E-03
Pu-241	1.9375E-04	1.8610E-04	1.8518E-04	1.8537E-04	1.8845E-04
Pu-242	2.9277E-04	2.8911E-04	2.8818E-04	2.9038E-04	2.9569E-04
Am-241	1.0791E-04	1.0465E-04	1.0353E-04	1.0686E-04	1.1421E-04
$Am\text{-}242^{m}$	9.2989E-06	9.0848E-06	9.0224E-06	9.1756E-06	9.4890E-06
Am-243	1.0017E-04	9.8324E-05	9.7993E-05	9.8630E-05	1.0032E-04
Cm-242	5.6250E-06	5.8208E-06	5.9476E-06	5.4901E-06	4.5416E-06
Cm-243	5.4321E-07	5.0246E-07	5.0136E-07	4.8876E-07	4.8480E-07
Cm-244	6.7240E-05	6.5722E-05	6.5622E-05	6.5349E-05	6.5394E-05
Cm-245	1.7397E-05	1.6743E-05	1.6663E-05	1.6696E-05	1.7026E-05
Cm-246	9.2285E-06	9.1426E-06	9.1307E-06	9.1364E-06	9.1805E-06
Zr	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03
[a] Mo	9.2873E-04	1.1464E-03	1.2031E-03	1.0625E-03	7.4065E-04

Table 3.8 Number Densities of Outer Core Fuel Pin, BOC (atoms/barn-cm)

Nuclide	Upper boundary from active core bottom (cm)				
	17.16	34.33	51.49	68.66	85.82
U-234	1.6317E-06	1.5766E-06	1.5638E-06	1.5894E-06	1.6552E-06
U-235	3.0822E-05	2.9870E-05	2.9561E-05	3.0391E-05	3.2250E-05
U-236	1.7881E-06	1.8534E-06	1.8941E-06	1.7528E-06	1.4710E-06
U-238	1.8244E-02	1.8144E-02	1.8115E-02	1.8191E-02	1.8359E-02
Np-237	9.8244E-05	9.7300E-05	9.6775E-05	9.8481E-05	1.0175E-04
Pu-236	7.1175E-10	8.2505E-10	8.4282E-10	8.0703E-10	6.8053E-10
Pu-238	1.6436E-04	1.6026E-04	1.5949E-04	1.6063E-04	1.6416E-04
Pu-239	2.8147E-03	2.7664E-03	2.7538E-03	2.7786E-03	2.8416E-03
Pu-240	1.7467E-03	1.7191E-03	1.7135E-03	1.7231E-03	1.7508E-03
Pu-241	2.8976E-04	2.8138E-04	2.8012E-04	2.8135E-04	2.8697E-04
Pu-242	4.0754E-04	4.0412E-04	4.0321E-04	4.0530E-04	4.1028E-04
Am-241	1.8607E-04	1.8127E-04	1.7970E-04	1.8397E-04	1.9339E-04
Am-242^{m}	1.2185E-05	1.2045E-05	1.2021E-05	1.2039E-05	1.2064E-05
Am-243	1.3234E-04	1.3019E-04	1.2985E-04	1.3036E-04	1.3206E-04
Cm-242	6.4688E-06	6.8630E-06	7.0553E-06	6.4446E-06	5.1976E-06
Cm-243	6.3471E-07	6.0893E-07	6.0901E-07	5.9753E-07	5.9372E-07
Cm-244	8.0107E-05	7.8889E-05	7.8847E-05	7.8479E-05	7.8359E-05
Cm-245	2.0200E-05	1.9678E-05	1.9613E-05	1.9635E-05	1.9913E-05
Cm-246	1.0443E-05	1.0371E-05	1.0361E-05	1.0367E-05	1.0410E-05
Zr	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03
[a)] Mo	8.1524E-04	1.0174E-03	1.0697E-03	9.4870E-04	6.6172E-04

Table 3.9 Number Densities of Inner Core Fuel Pin, EOC (atoms/barn-cm)

Nuclide	Upper boundary from active core bottom (cm)				
	17.16	34.33	51.49	68.66	85.82
U-234	1.8287E-06	1.7464E-06	1.7210E-06	1.7764E-06	1.9079E-06
U-235	2.4267E-05	2.2703E-05	2.2106E-05	2.3630E-05	2.7069E-05
U-236	3.8348E-06	3.8253E-06	3.8904E-06	3.6660E-06	3.2386E-06
U-238	1.9080E-02	1.8846E-02	1.8774E-02	1.8956E-02	1.9348E-02
Np-237	4.2039E-05	4.2589E-05	4.2264E-05	4.3414E-05	4.5039E-05
Pu-236	5.8848E-10	7.1975E-10	7.3569E-10	7.1148E-10	5.8989E-10
Pu-238	1.1306E-04	1.0695E-04	1.0560E-04	1.0811E-04	1.1477E-04
Pu-239	2.1915E-03	2.1613E-03	2.1525E-03	2.1639E-03	2.1956E-03
Pu-240	1.3132E-03	1.2652E-03	1.2570E-03	1.2692E-03	1.3112E-03
Pu-241	1.9156E-04	1.8154E-04	1.8043E-04	1.8083E-04	1.8570E-04
Pu-242	2.8455E-04	2.7834E-04	2.7677E-04	2.8014E-04	2.8850E-04
Am-241	9.5712E-05	9.0753E-05	8.8828E-05	9.3685E-05	1.0442E-04
Am-242^{m}	9.4584E-06	9.0121E-06	8.9023E-06	9.1452E-06	9.7079E-06
Am-243	9.4543E-05	9.1729E-05	9.1054E-05	9.2383E-05	9.5601E-05
Cm-242	7.0997E-06	7.2237E-06	7.4075E-06	6.9780E-06	6.1283E-06
Cm-243	7.7326E-07	7.5394E-07	7.7307E-07	7.1317E-07	6.3410E-07
Cm-244	7.2896E-05	7.0894E-05	7.1010E-05	7.0175E-05	6.9521E-05
Cm-245	1.6466E-05	1.5274E-05	1.5116E-05	1.5268E-05	1.6039E-05
Cm-246	9.3179E-06	9.1820E-06	9.1659E-06	9.1722E-06	9.2397E-06
Zr	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03
[a)] Mo	2.1157E-03	2.5912E-03	2.7287E-03	2.4279E-03	1.7425E-03

Table 3.10 Number Densities of Outer Core Fuel Pin, EOC (atoms/barn-cm)

Nuclide	Upper boundary from active core bottom (cm)				
	17.16	34.33	51.49	68.66	85.82
U-234	2.6749E-06	2.5827E-06	2.5566E-06	2.6029E-06	2.7159E-06
U-235	2.6653E-05	2.5248E-05	2.4731E-05	2.5879E-05	2.8390E-05
U-236	2.8172E-06	2.8830E-06	2.9516E-06	2.7665E-06	2.4368E-06
U-238	1.7914E-02	1.7750E-02	1.7697E-02	1.7813E-02	1.8063E-02
Np-237	8.9603E-05	8.8307E-05	8.7362E-05	8.9756E-05	9.3984E-05
Pu-236	9.8954E-10	1.2250E-09	1.2619E-09	1.2075E-09	9.6571E-10
Pu-238	1.6477E-04	1.5854E-04	1.5736E-04	1.5898E-04	1.6427E-04
Pu-239	2.7012E-03	2.6334E-03	2.6132E-03	2.6495E-03	2.7397E-03
Pu-240	1.7293E-03	1.6833E-03	1.6741E-03	1.6886E-03	1.7341E-03
Pu-241	2.7597E-04	2.6534E-04	2.6364E-04	2.6552E-04	2.7377E-04
Pu-242	4.0028E-04	3.9414E-04	3.9250E-04	3.9569E-04	4.0365E-04
Am-241	1.7437E-04	1.6693E-04	1.6412E-04	1.7032E-04	1.8364E-04
Am-242^{m}	1.3196E-05	1.2874E-05	1.2832E-05	1.2894E-05	1.3097E-05
Am-243	1.2765E-04	1.2462E-04	1.2395E-04	1.2502E-04	1.2795E-04
Cm-242	8.2990E-06	8.6904E-06	9.0040E-06	8.4227E-06	7.4190E-06
Cm-243	8.0947E-07	8.1124E-07	8.3343E-07	7.7944E-07	7.1488E-07
Cm-244	8.4614E-05	8.3114E-05	8.3342E-05	8.2455E-05	8.1924E-05
Cm-245	1.9441E-05	1.8483E-05	1.8342E-05	1.8467E-05	1.9145E-05
Cm-246	1.0499E-05	1.0391E-05	1.0377E-05	1.0385E-05	1.0451E-05
Zr	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03	7.2802E-03
[a)] Mo	1.8140E-03	2.2465E-03	2.3733E-03	2.1324E-03	1.5594E-03

The remaining non-fuel materials used are described in the following table:

Table 3.11 Number Densities of Coolant and Structural Materials of ABR Metallic Core (atoms/barn-cm)

Material	Nuclide	Number density
Lower structure (homogeneous mixture of SS-316 and Sodium)	Na	1.5591E-02
	Fe	1.5878E-02
	Ni	3.2604E-03
	Cr	3.2355E-03
	Mn-55	5.0846E-04
	Mo	4.3524E-04
Coolant	Na	2.2272E-02
HT-9	Fe	6.9715E-02
	Ni	4.2984E-04
	Cr	1.0366E-02
	Mn-55	4.5921E-04
	Mo	4.9007E-04
Natural B_4C	C	1.9657E-02
	B-10	1.5018E-02
	B-11	6.3609E-02
Enriched B_4C	C	2.0632E-02
	B-10	5.3642E-02
	B-11	2.8884E-02

CHAPTER 4. SERPENT MODEL VERIFICATION

The goal of this thesis involves developing a process of coupling Monte Carlo and nodal diffusion methods for core analysis with results comparable to that of full Monte Carlo, while saving time and computational resources. Therefore, in order to obtain a full core Monte Carlo reference solution, the first step taken was to model the ABR core using Serpent. To verify the accuracy of the Serpent model, results are compared to those reported in the benchmark [2] using both BoC and EoC fuel compositions. The benchmark's main purpose was to quantify and understand the uncertainties related to modelling steady-state and transient calculations of Generation IV reactors. In total, 11 institutions from 7 countries contributed 31 sets of results using a variety of data libraries, computational methods and reactor physics codes, including MCNP5, ERANOS VARIANT8, VARIANT11, TRIPOLI-4, and Serpent, to name a few.

4.1 Serpent Model

The full core 3D ABR model was built using Serpent 2. Serpent utilizes the universe-based constructive solid geometry model [13], which allows for virtually any 2D and 3D geometry to be constructed. The way a reactor core model is built in Serpent can be viewed as an "inside-out" approach. First, all materials and surfaces used in the model must be defined. The user then proceeds by defining the geometry of the inner most division of the reactor, typically the fuel (in the case of this ABR model, fuel pins surrounded by cladding are first defined). The fuel pins are then arranged into the next level of the geometry, which are hexagonal prism assemblies in the ABR core. Once all assemblies have been properly defined, they are arranged into a new universe, representing

the full core. Finally, the outermost boundary of the geometry must be defined by specifying what is referred to as “outside cell”, at which point the pre-defined boundary conditions are applied.

The following images were generated using Serpent’s plotting feature, and show the radial cross-sectional view of the core (with control rod assemblies inserted for illustrative purposes), as well as an axial view of the full core 3D model (with control rods removed, as is the case during nominal operation).

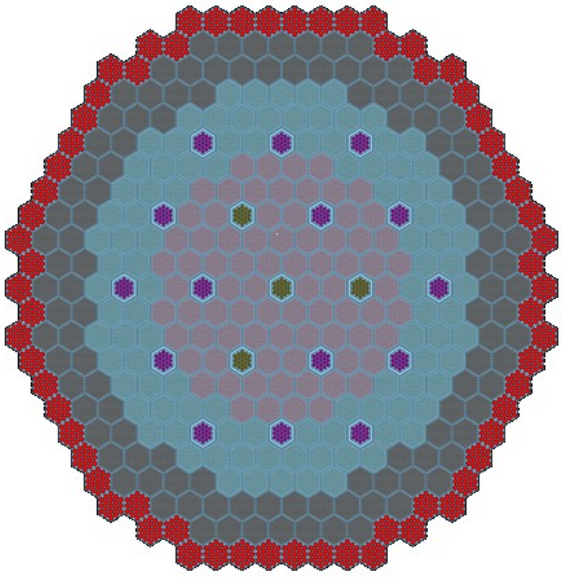


Figure 4.1 Radial View of ABR Core Model Built in Serpent

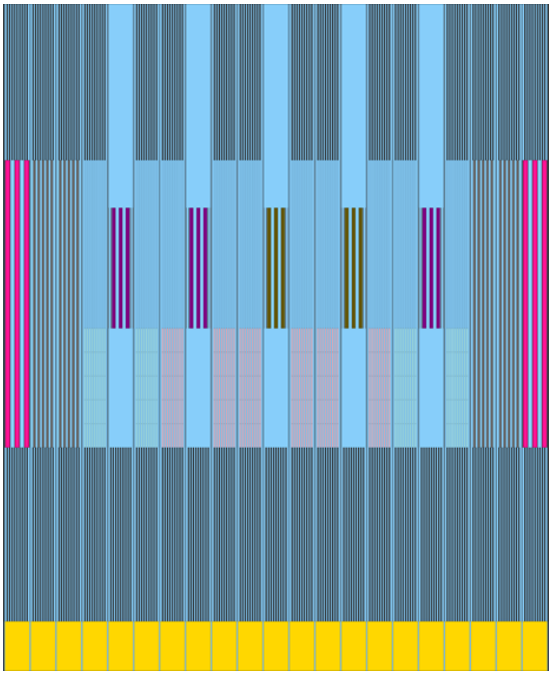


Figure 4.2 Axial View of ABR Serpent 3D Model

4.2 Comparison to Benchmark Results

In order to establish the validity of this ABR 3D full core model to be used as a reference solution, various static state calculations were performed to meet the scenarios described in the benchmark exercises, which included:

- Core multiplication factor (k_{eff}).
- Sodium void worth – defined by the change in reactivity $\Delta\rho = \rho_{void} - \rho_{nominal}$, where ρ_{void} in this exercise is defined by voiding all sodium present in the core. This is accomplished in Serpent by reducing the sodium density to 0.1% that of the nominal value.

- Doppler constant – defined by $K_D = \frac{\rho_{high} - \rho_{nominal}}{\ln 2}$, where the subscript high stands for the core reactivity when the fuel temperature is raised by a factor of two to that of the nominal value.
- Effective delayed neutron fraction (β_{eff}).
- Control rod worth – defined as the change in reactivity $CR_{worth} = \rho_{CR} - \rho_{nominal}$, where ρ_{CR} is the core reactivity when all control assemblies (both primary and secondary) are fully inserted in the core.

The runs used the ENDF/B-VII cross section library [14]. Steady state calculations using 500,000 neutron histories, 1000 active cycles and 200 inactive cycles for each of the required scenarios were performed at the BoC and EoC. A summary of the results is presented:

Table 4.1 Summary of Results Comparing Serpent Model to Benchmark Averages

	k-eff	β_{eff}	$\Delta\rho$ Sodium	$\Delta\rho$ Doppler	$\Delta\rho$ Sodium
	Beginning of Cycle				
Benchmark Average (B.A.)	1.0355	345	2024	-347	19647
$\pm$ Standard Deviation	0.0078	10	407	44	2087
Serpent Model (uncertainity)	1.0355 (7E-5)	338	2033	-303	17878
Difference from B.A.	0.0020	7	-9	-44	1799
	End of Cycle				
Benchmark Average (B.A.)	1.0123	344	2146	-348	20497
$\pm$ Standard Deviation	0.0071	12	435	36	2228
Serpent Model (uncertainity)	1.0196 (7E-5)	335	2068	-314	18597
Difference from B.A.	-0.0073	9	78	-34	1900

4.3 Discussion of Results

The results obtained from the Serpent runs are compared to the average benchmark results reported. As seen in Table 4.1, very good agreement is achieved when comparing the 3D full core ABR model to the results reported by other benchmark participants. Most results are within 1 standard deviation from the reported average, and all results compared are well within 2 standard deviations. The values for k_{eff} and β_{eff} are provided directly by the Serpent output file. The results corresponding to changes in reactivity take the

reactivity obtained from the nominal run and calculate the different values in accordance to the benchmark description and defined in the previous section.

These results indicated that the Serpent model built in a manner such that the physical properties of the reactor are modeled well and represent a valid reference solution to be used in the remaining of the work performed in this thesis. The focus is now shifted towards applying the Serpent-DYN3D code sequence for the analysis of the ABR core.

CHAPTER 5. SERPENT-DYN3D COMPARISON

This chapter will describe the initial comparisons made between Serpent and DYN3D. Some work was performed with very basic models of fuel pins and 2D lattices in order to gain expertise with the DYN3D code and the cross-section generation process. However, this work is not reported as it is very elementary and does not add much insight into the Serpent-DYN3D application process. Therefore, we start by comparing DYN3D's burnup and power distribution calculation capabilities to that of Serpent's reference solutions for realistic cases of 2D fuel assemblies and 3D full core. The first step towards applying the code sequence is to generate homogenized few-group constants.

5.1 Description of DYN3D Model

As described in Chapter 2, DYN3D is a nodal diffusion reactor dynamics code. The input of the code is provided by sub-dividing the reactor into nodes. These nodes are typically in the order of 20 cm in height, but each layer can be assigned a specific length. Each node is described as being composed of a single and unique cross-section set, which in reality is a homogenous mixture of the materials included where the cross sections were generated. Associated with each material is a cross section input file, which includes the following parameters (** indicates values that are associated with each energy group used)*

- Transport cross section*;
- Absorption cross section*;
- Fission-neutron production cross section*;
- Fission energy-production cross section*;
- Scattering matrix*;

- Assembly Discontinuity Factor (optional)*;
- Delayed-neutron Fractions;
- Delayed-neutron precursor decay constants;
- Inverse neutron velocity*;
- Fission spectrum*

These values must be provided for each of the burnup steps in which the cross sections were generated. A sample 3D assembly is presented with the associated input files for each material (the ABR model in DYN3D is built using hexagonal nodes, the following figure is simply for illustrative reasons).

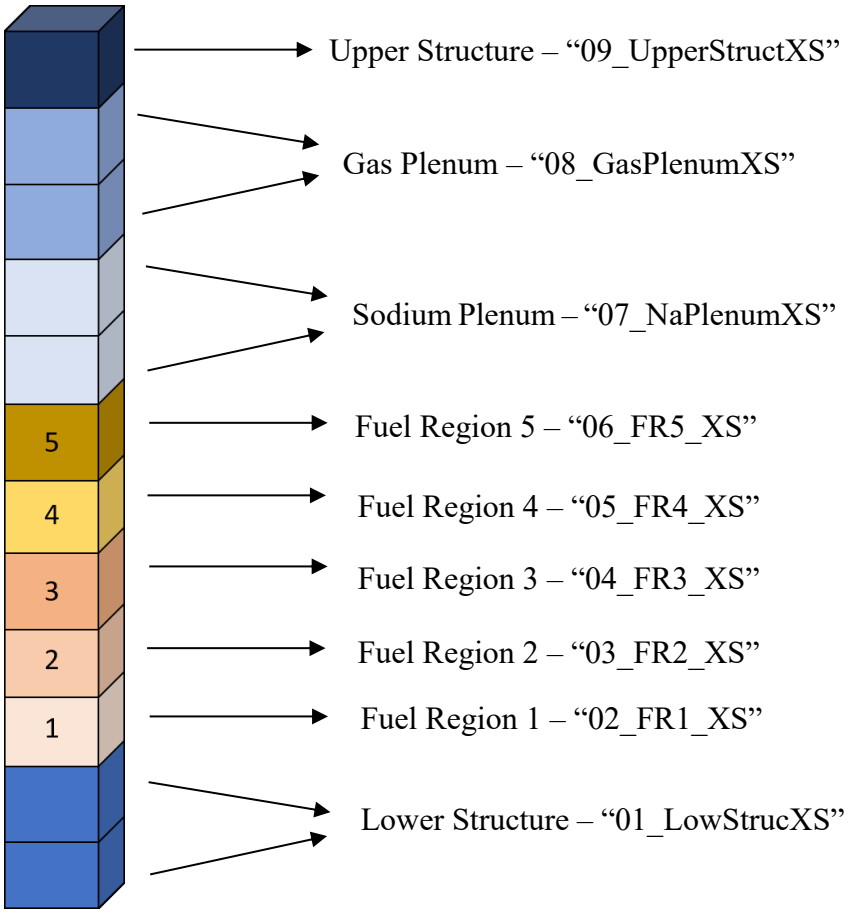


Figure 5.1 Example of Nodal Structure for Fuel Assembly in DYN3D

The process used for generating such cross-section files is described in the following section.

5.2 Cross-Section Generation

The generation of cross sections can be considered the most important step in the coupling of the Serpent and DYN3D codes, as the accuracy of the results generated in DYN3D are highly dependent on the quality of group constants provided. For this initial analysis, only macroscopic cross sections are used. The process used to generate microscopic cross section will be further explained in the next chapter. Throughout the chapter, the generation of infinite spectrum (*INF*) and B1 leakage corrected (*B1*) cross section will be performed and the effect that each approach has on the calculations will be analyzed.

Throughout the work in this thesis, a 24-group energy structured was used. The selected group boundaries are presented in the following table:

Table 5.1 Energy Group Structure

Group number	Upper energy limit (MeV)	Group number	Upper energy limit (MeV)	Group number	Upper energy limit (MeV)
1	1.0000E+01	9	1.8316E-01	17	3.3546E-03
2	6.0653E+00	10	1.1109E-01	18	2.0347E-03
3	3.6788E+00	11	6.7380E-02	19	1.2341E-03
4	2.2313E+00	12	4.0868E-02	20	7.4852E-04
5	1.3534E+00	13	2.4788E-02	21	4.5400E-04
6	8.2085E-01	14	1.5034E-02	22	3.1203E-04
7	4.9787E-01	15	9.1188E-03	23	1.4894E-04
8	3.0197E-01	16	5.5309E-03	24	1.0000E-10

The choice of energy structure is based on the study presented by Fridman et al., "Modeling of SFR cores with Serpent-DYN3D codes sequence", where it was found that using the ECCO 33-group energy structure resulted in too few events being captured at the most thermal groups, leading to high statistical uncertainty [7]. The solution found was to lump some of the lower energy groups together, therefore increasing the probability that events are captured in this region, and decreasing the overall number of energy groups used.

5.2.1 2D Fuel Lattice

We start by modelling a 2D ABR fuel lattice which is infinite in the axial direction. In Serpent, this is achieved by modelling a single fuel assembly and using reflective boundary conditions, as to represent an assembly surrounded by identical assemblies, as seen in the following figure:

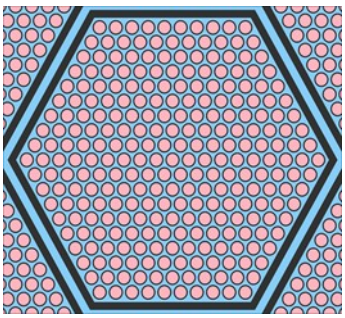


Figure 5.2 Single 2D Fuel Assembly with Reflective Boundary Conditions

Serpent generates the few group constants based on assigned universes, as described in Chapter 4. Therefore, for this assembly, everything inside of the assembly duct is defined as a universe, and Serpent homogenizes the materials within the universe to generate the cross sections. For this analysis, the assembly was depleted under normal operating conditions, with a fuel temperature of 807 K and a coolant density of

2.2272E-02 atoms/barn-cm, and the 24-group structure presented in Table 5.1 is utilized. Since the actual ABR assembly has fuel composition varying axially and this is only a 2D model, for sake of completeness each fuel region was modeled and compared to DYN3D results.

5.2.2 3D Fuel Assembly

While Serpent allows for the construction of complex and highly heterogeneous geometries, DYN3D input requires a single set of cross sections for a given node. These cross sections are homogenized in Serpent over a given geometry. To generate cross sections for this study, the following approach was used:

- The cross sections for fuel regions were generated using a 3D single assembly model of a fuel assembly with reflective boundary conditions radially and black boundary conditions axially. The model simultaneously generates group constants for each of the axial fuel layers, as depicted in Figure 5.3. The fuel layer universes in Serpent are built with the intent of replicating the exact dimensions of the corresponding node in DYN3D.
- All group constants for non-multiplying regions are separately generated using 2D super-cell models [15], as depicted in Figure 5.4. The regions of interest are placed in the center of the model and surrounded by fuel assemblies to simulate the flux similar as to the flux the assembly would experience in a full core. The group constants are only homogenized over the central region of interest.

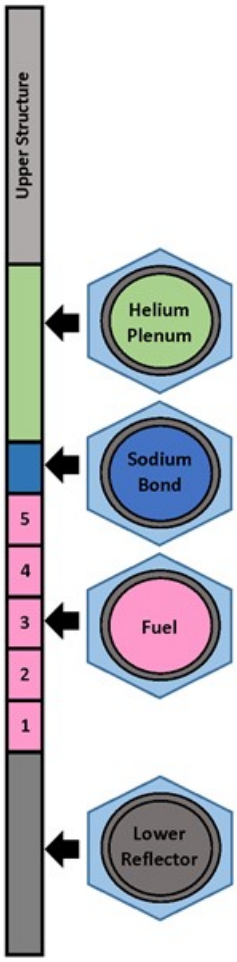


Figure 5.3 Axial Representation of Fuel Regions for Cross Section Generation

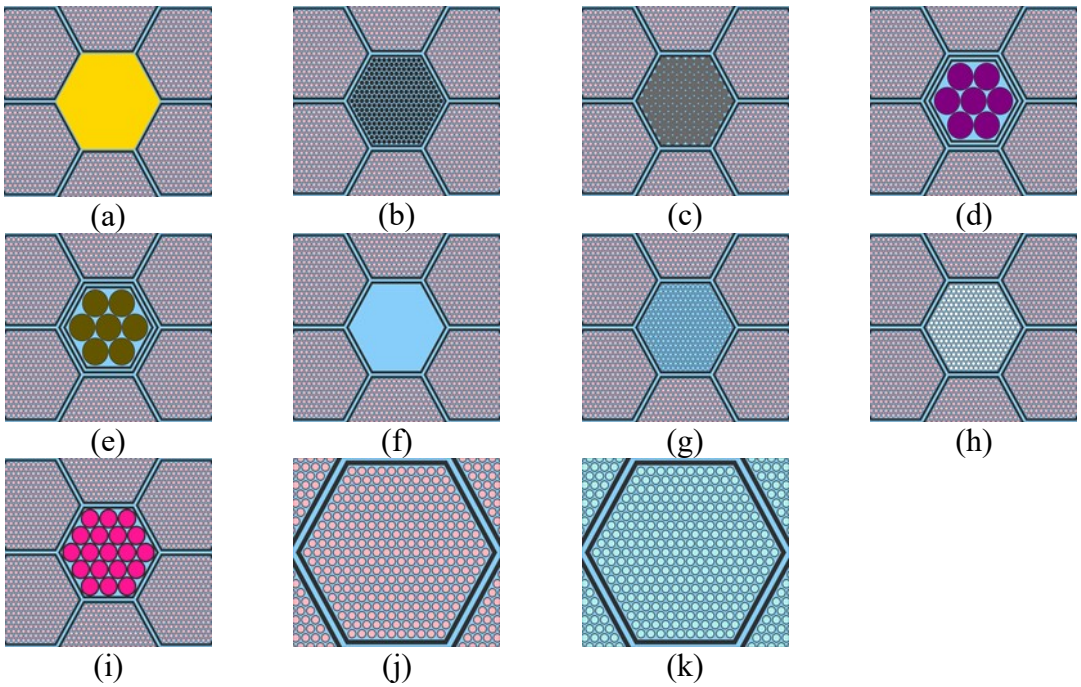


Figure 5.4 Individual Sections Supercell Models: (a) Lower Structure, (b) Lower Reflector and Upper Structure, (c) Radial Reflector, (d) Primary Control Assembly, (e) Secondary Control Assembly, (f) Empty Duct, (g) Sodium Plenum, (h) Helium Gas Plenum, (i) Radial Shield, (j) Inner Core Fuel Assembly, (k) Outer Core Fuel Assembly

Once the set of cross sections for each material is generated, the DYN3D full core model can be built by assigning the adequate material to each node. The comparison of the Serpent-DYN3D to the reference Serpent solution can now be performed.

5.3 Results

5.3.1 2D Fuel Lattice

The 2D infinite fuel lattice describe in Section 5.2.1 is modeled in DYN3D and Serpent and results for the lattice depletion are presented in Figure 5.5, in which the DYN3D calculation, performed twice, once using *B1* cross sections and another using infinite spectrum cross sections, are compared to the Serpent results:

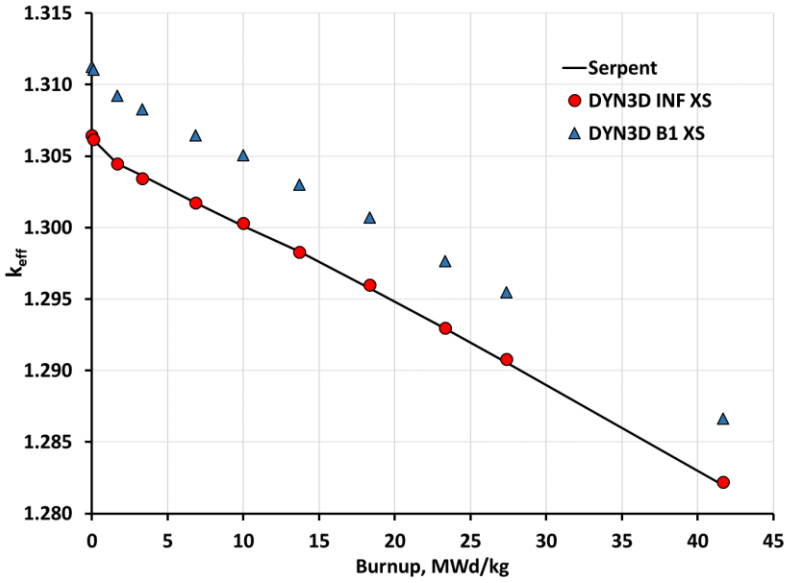


Figure 5.5 Single 2D Fuel Lattice Burnup Calculation Results for Serpent vs. DYN3D

The results in terms of change in reactivity, calculated as $\Delta\rho = 10^5 * \left(\frac{1}{k_{Serpent}} - \frac{1}{k_{DYN3D}}\right)$, where Serpent is taken as the reference solution, are also compared throughout the depletion cycle, where 2σ refer to two standard deviations from the Serpent reference solution:

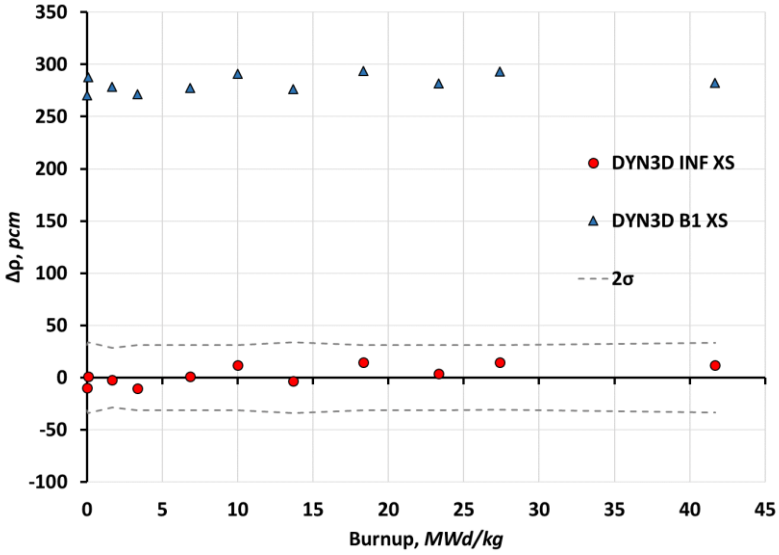


Figure 5.6 Single Assembly Difference in Reactivity for Serpent vs. DYN3D

As previously mentioned, this initial comparison uses the fuel composition of the first layer on the inner core. Since this is a 2D model, for the sake of completeness and in order to verify consistency, the material of each fuel layer of the inner and outer core is also analyzed.

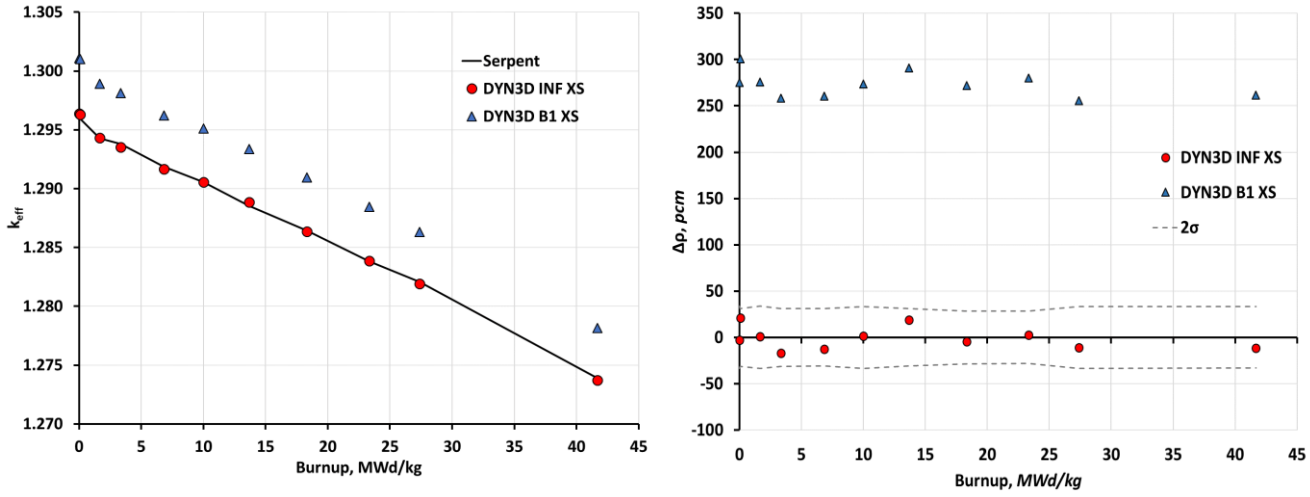


Figure 5.7 Results for Inner Core, Region 2

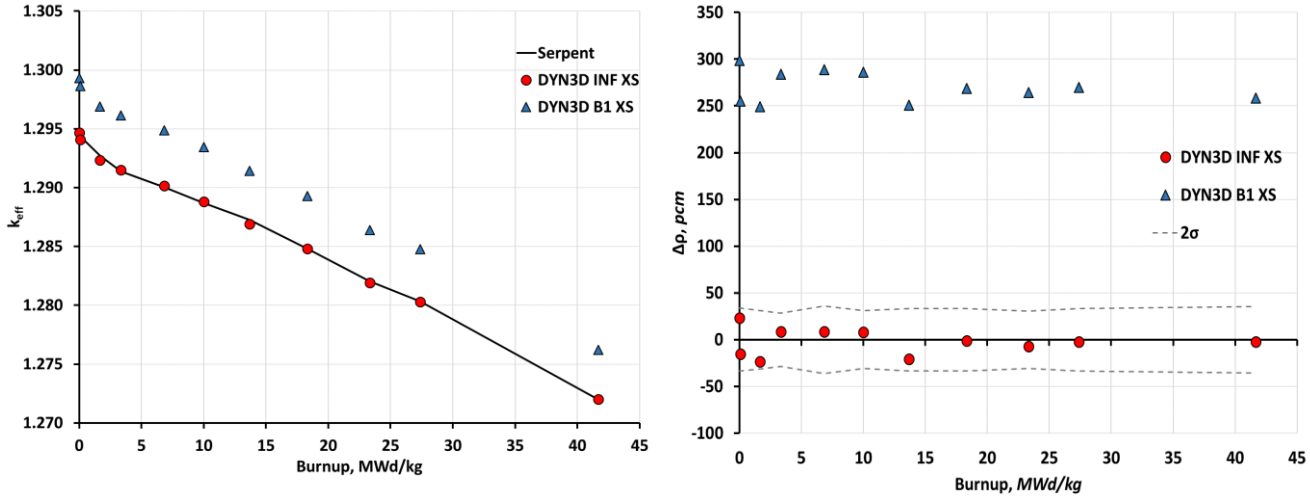


Figure 5.8 Results for Inner Core, Region 3

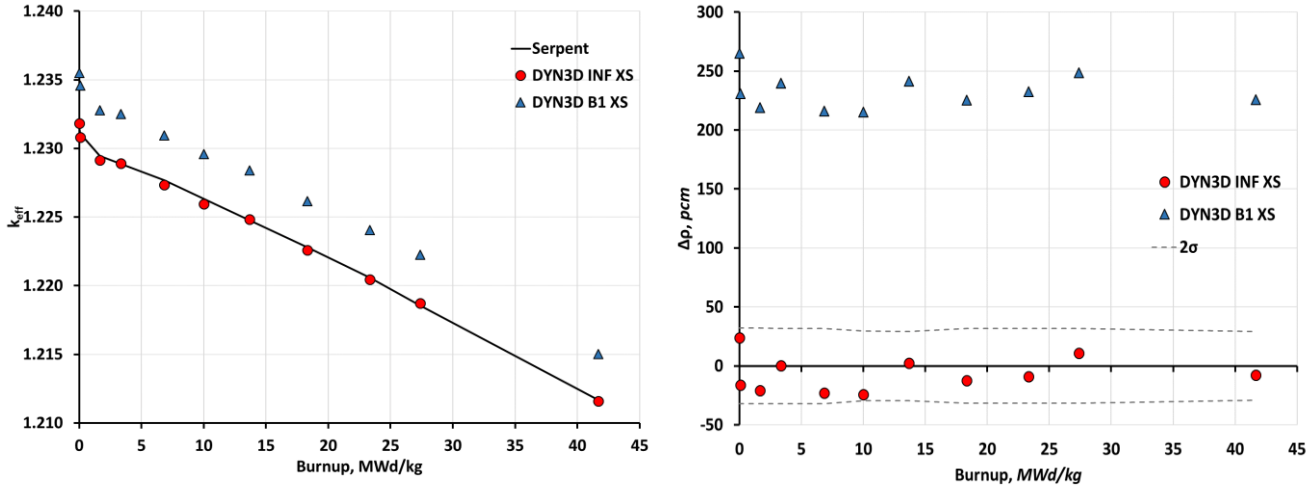


Figure 5.9 Results for Inner Core, Region 4

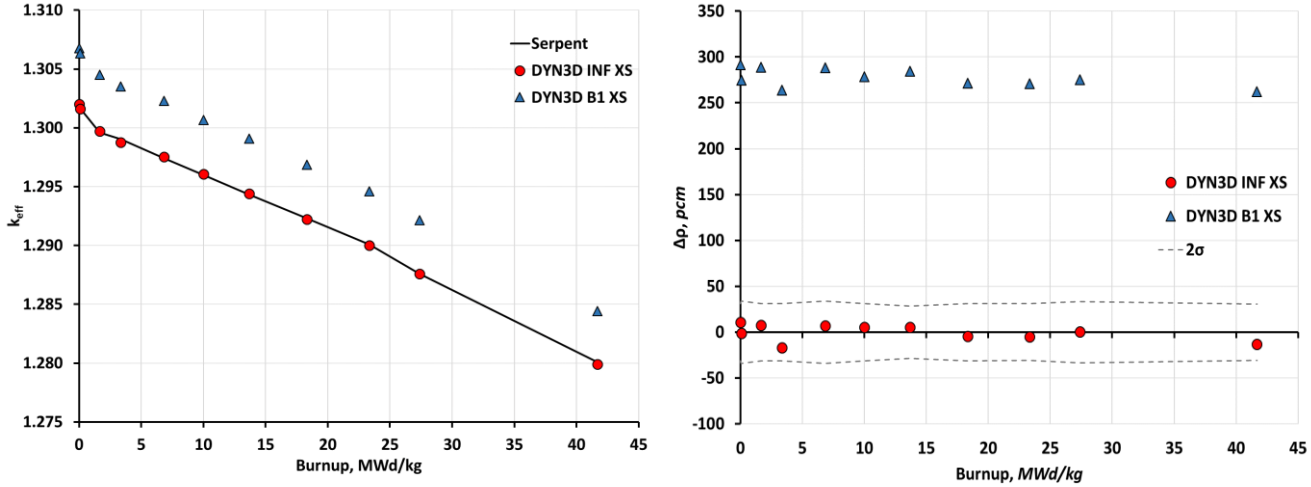


Figure 5.10 Results for Inner Core, Region 5

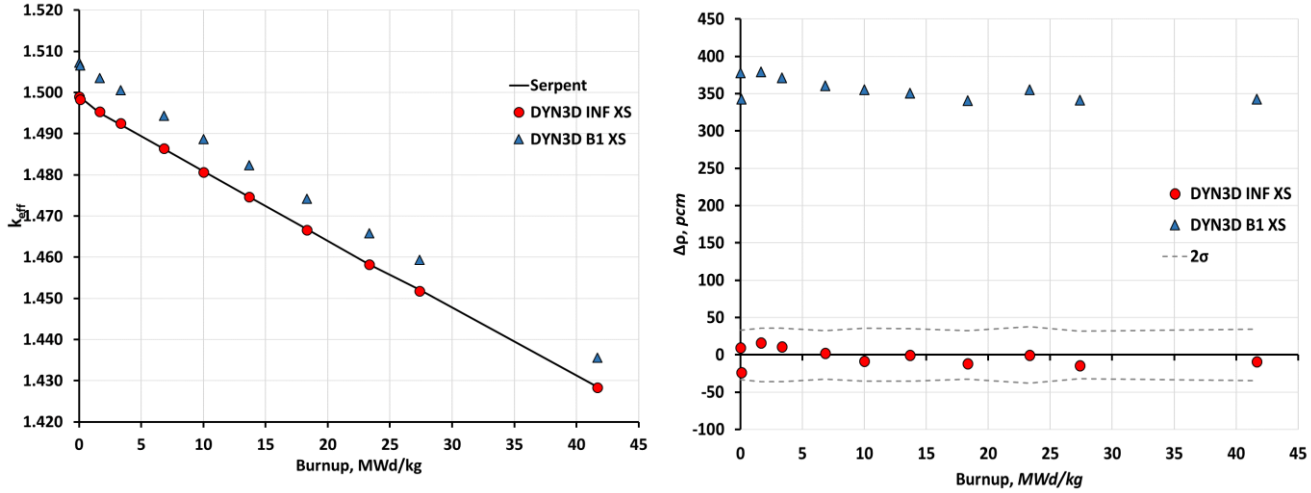


Figure 5.11 Results for Outer Core, Region 1

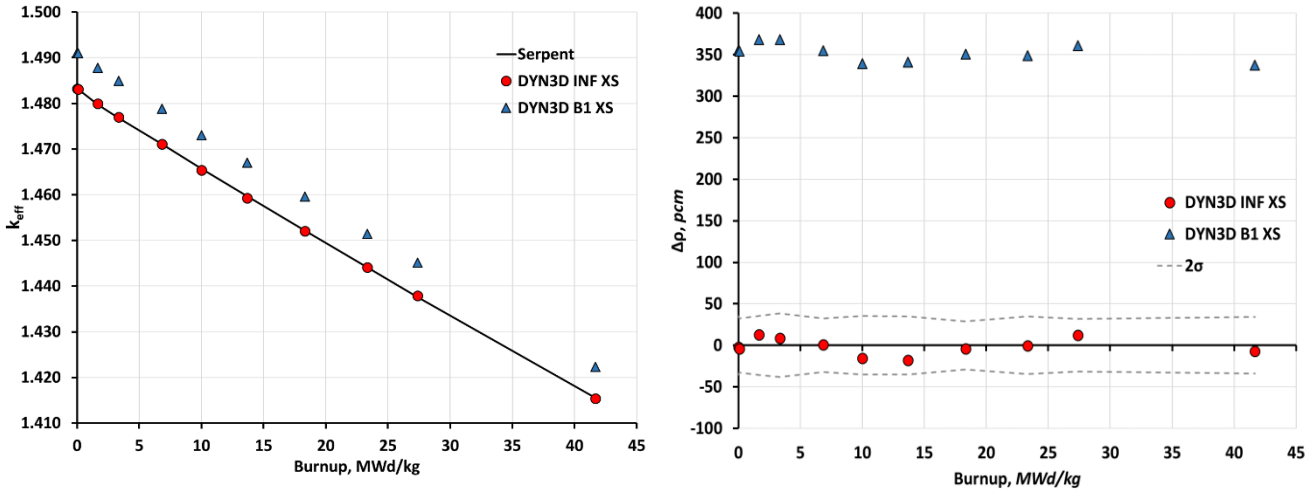


Figure 5.12 Results for Outer Core, Region 2

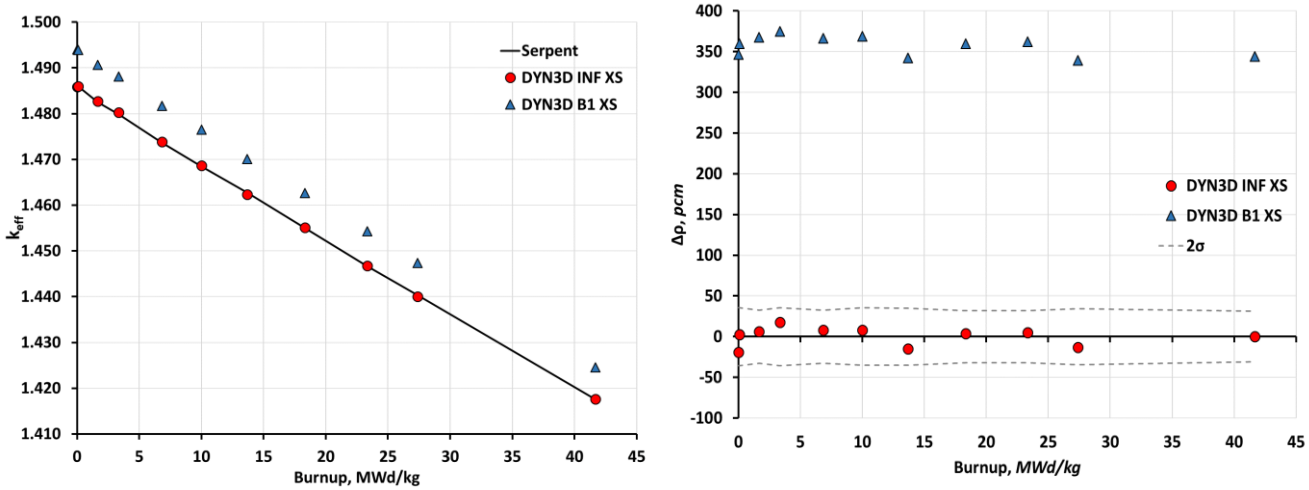


Figure 5.13 Results for Outer Core, Region 3

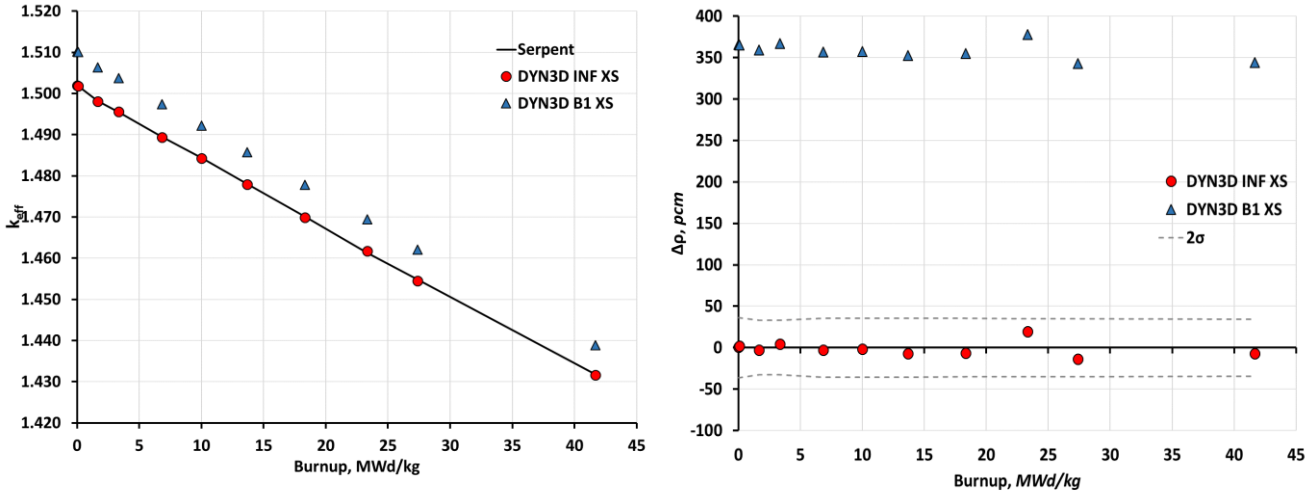


Figure 5.14 Results for Outer Core, Region 4

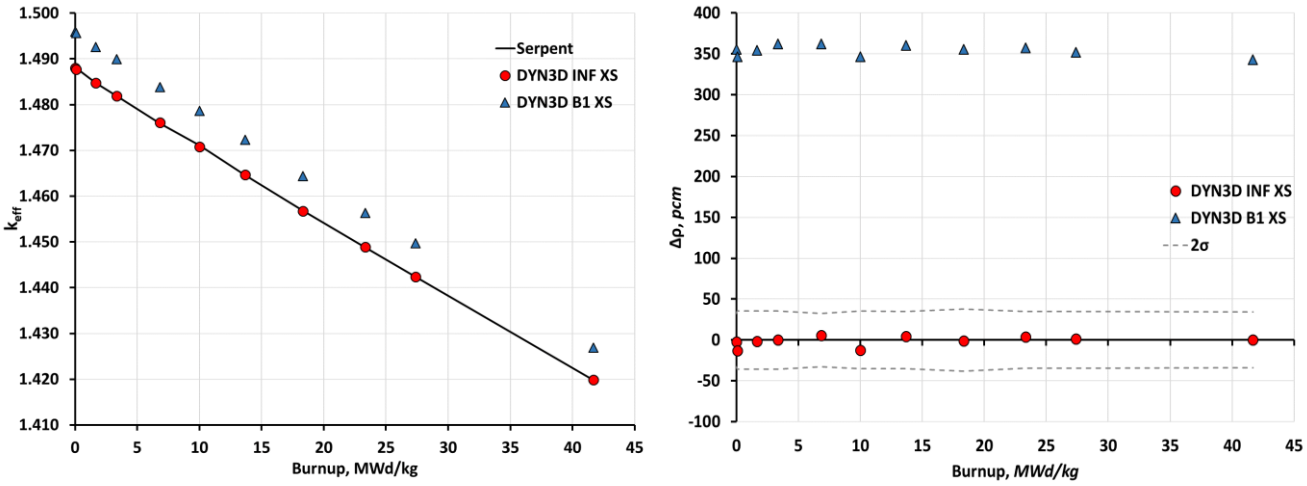


Figure 5.15 Results for Outer Core, Region 5

It can be seen that for an infinite system, in which there is no neutron leakage, the *INF* group constants, as expected, are the appropriate set of cross sections to be used, as the results obtained from DYN3D are well within the statistical error reported by Serpent. The *B1* cross sections generate a discrepancy in the order of 300 pcm throughout the cycle. Therefore, the Serpent-DYN3D sequence is shown to be capable of accurately performing the depletion analysis in this infinite fuel lattice model.

5.3.2 3D Full Core

Once the 2D model was shown to reproduce the reference solution quite well, the full core was modeled in 3D. For this case, the fuel assemblies of the inner and outer core (as shown in Figure 5.3) were depleted for a full cycle, with the cross sections for each axial fuel region being generated for each burnup step. The cross sections for non-multiplying regions were assumed to remain constant throughout the cycle. Once again, the *B1* and *INF* cross sections were generated and applied for comparison to the reference solution, as shown in Figure 5.16.

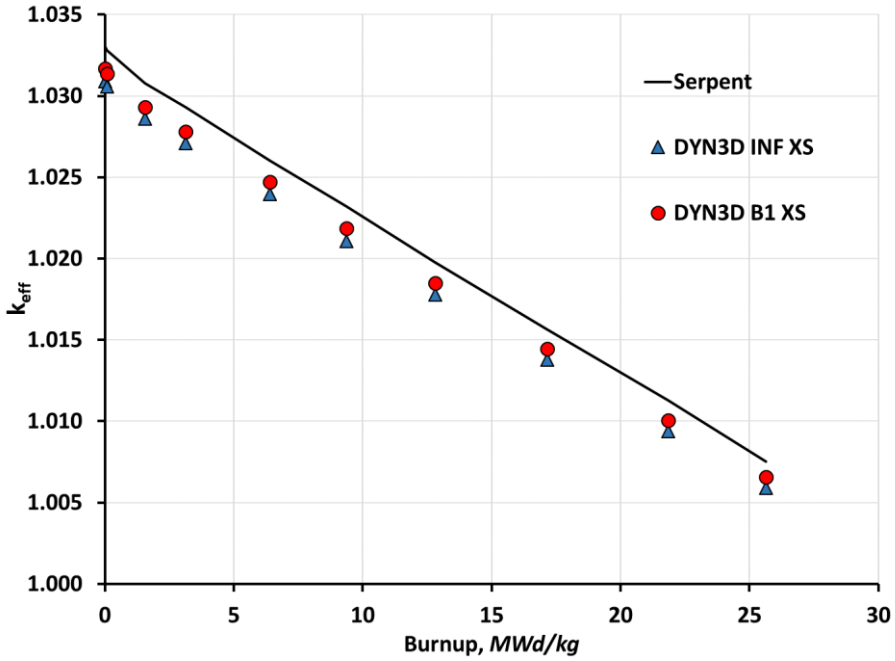


Figure 5.16 Full Core Burnup Calculation Results for Serpent vs. DYN3D

Comparing the change in reactivity for full core is not as meaningful as for the 2D case, since larger discrepancy is to be expected, of which would be well beyond the statistical error reported by Serpent. Instead, the difference in radial power distribution reported by both codes is compared and plotted for one quarter of the core.

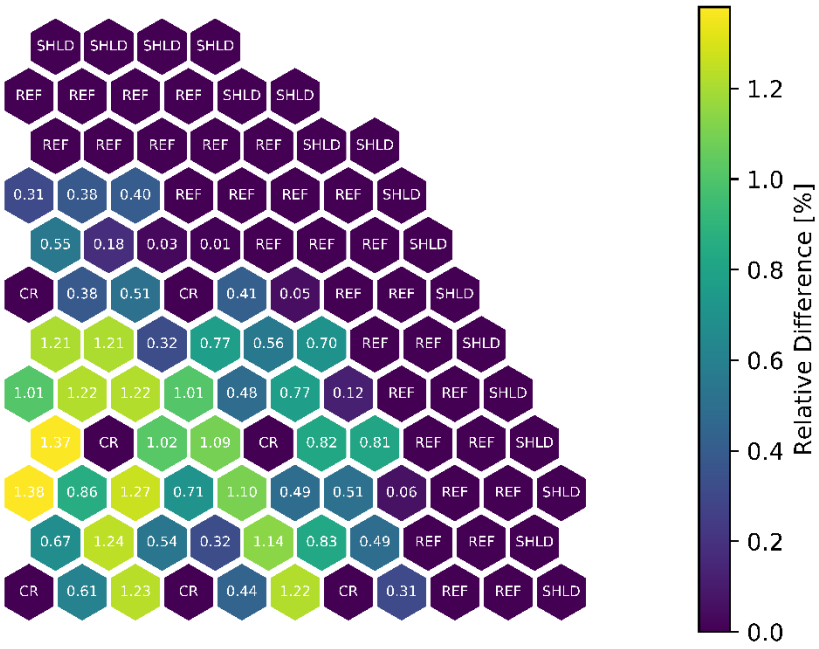


Figure 5.17 Difference in Radial Power Distribution at Beginning of Cycle

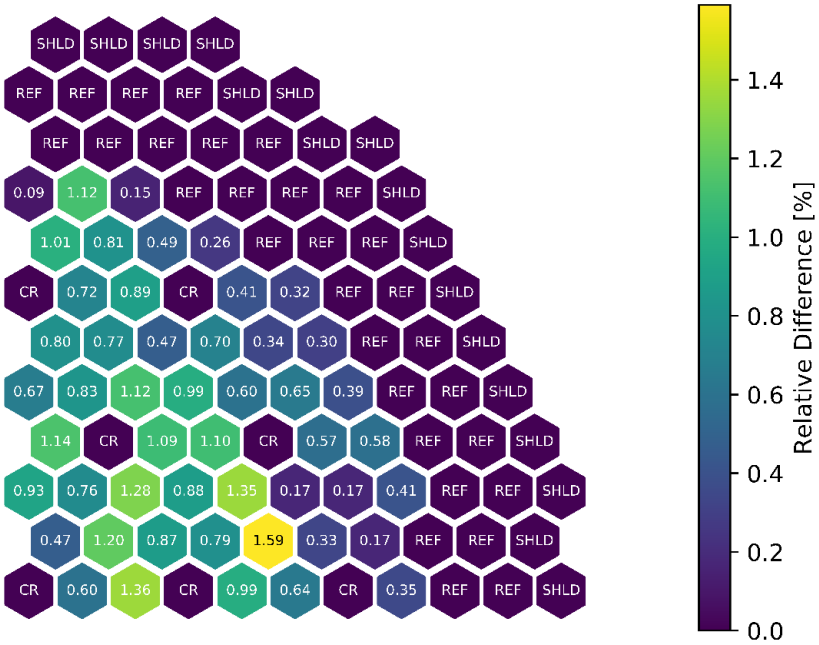


Figure 5.18 Difference in Radial Power Distribution After 1 Day

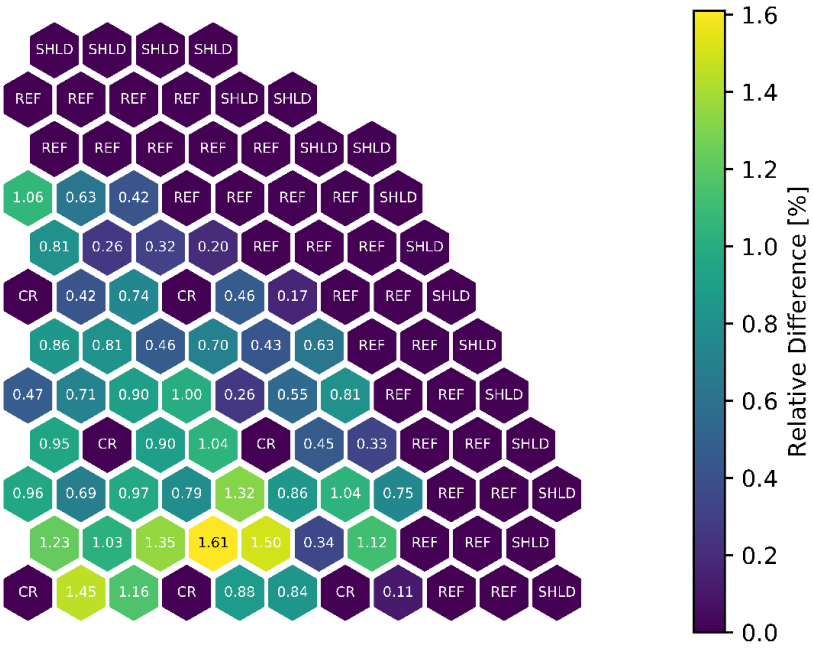


Figure 5.19 Difference in Radial Power Distribution After 20 Days

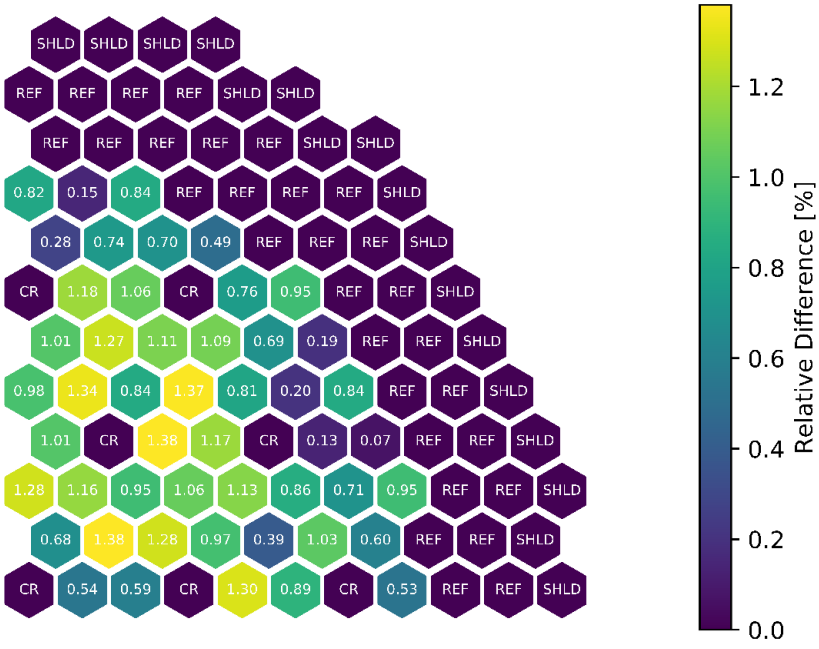


Figure 5.20 Difference in Radial Power Distribution After 160 Days

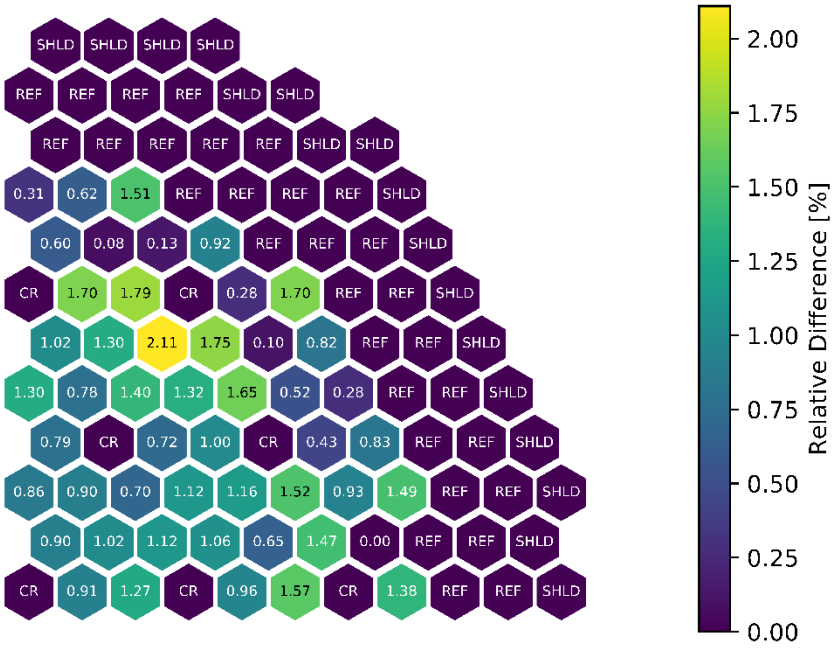


Figure 5.21 Difference in Radial Power Distribution at the End of Cycle (328.5 Days)

In Figure 5.16, the results for the full core depletion analysis are shown, comparing the Serpent full core 3D model to the results obtained using the Serpent-DYN3D sequence, where DYN3D calculations done using both *INF* and *B1* group constants (for fuel regions only) are plotted. Contrary to the 2D assembly case, in the full core model, where the system is in its critical state and the spectrum should be corrected, the results indicate that using the *B1* group constants leads to better agreement with the Serpent reference (124 pcm average reactivity difference) than the infinite spectrum (194 pcm average reactivity difference). It can be concluded that applying the leakage correction to the group constants produce better results when compared to the reference case.

For the radial power distribution comparison, the full core was depleted with only 5 burnup steps to save on computational time, as adding the detectors necessary for obtaining the power distribution add computational cost to the Monte Carlo calculation.

The plots presented in Figure 5.17 through Figure 5.21 present the average power difference calculated by each code in each assembly, at BoC, EoC and intermediate burnup steps. The highest difference in axial average power at the BoC was 1.38%, with an RMS error 8.6E-03. At the EoC comparison, the highest difference in average axial power reported is 2.11%, with an RMS error of 1.17E-02. Although there is a slightly higher average difference at the end of cycle, there is still a very good agreement between the two codes investigated. The following table summarizes the full core results for arbitrary sampled time-points:

Table 5.2 Summary of Full Core Depletion Calculation Results.

BU, MWd/kg	BU, Days	k_{eff} DYN3D	k_{eff} Serpent	MC Error	Relative Difference	Radial Power MSR Error	Maximum Radial Error (%)
0	0	1.031665	1.03307	4.70E-05	136.0	8.60E-03	1.38
0.078	1	1.031360	1.03286	4.50E-05	145.2	8.32E-03	1.59
1.560	20	1.029317	1.03081	4.60E-05	144.8	8.81E-03	1.61
12.481	160	1.018813	1.02024	4.40E-05	139.9	9.69E-03	1.38
25.625	328.5	1.006555	1.00768	4.80E-05	111.6	1.17E-02	2.11

5.4 Serpent-DYN3D Comparison Conclusions

Depletion analysis of the ABR reactor was performed using the Serpent-DYN3D code sequence. While Serpent models were used to generate reference solutions, the supercell method was utilized to produce few group constants to be used in DYN3D. The Serpent-DYN3D code sequence is compared to reference Serpent models for individual assemblies as well as for a 3D full core model. Results in reactivity and radial power distribution show very good agreement, indicating that this code sequence can be used to model depletion of full core metal fueled fast reactors. Finally, the Serpent-DYN3D code sequence is applied for decay heat analysis.

CHAPTER 6. DECAY HEAT ANALYSIS

The ability to determine the decay heat produced by nuclear fuel after reactor shutdown is of extreme importance for analyses of accident scenarios in which heat removal systems might be compromised. It is also important for designing storage and transportation solutions for spent fuel. In Light Water Reactors (LWRs), the decay heat generated by the fuel after shutdown can be predicted with the help of the decay heat standards such as American National Standard for Decay Heat Power in Light Water Reactors issued by American Nuclear Society [16]. However, in the case of advanced reactors with non-conventional fuels, using such standardized calculation procedure can lead to large discrepancies in decay heat calculations. Alternative approaches must be capable of appropriately accounting for fuel composition and neutron energy spectrum [17]. First, this chapter investigates the effect that operational conditions have on decay heat. The remainder of the chapter presents a method for decay heat analysis applying the Serpent-DYN3D code sequence for 2D lattice, 3D fuel assembly and full core 3D models.

6.1 Preparatory Work for Decay Heat Analysis

6.1.1 Effect of Operational Conditions on Decay Heat

The first step taken is to analyze the effects that operational conditions have on decay heat generation. This is accomplished by performing depletion analysis and using the isotopic atomic density in different points in time to calculate decay heat. To perform the depletion analysis, Serpent 2 was used. Serpent provides a detailed list of isotopic composition at each of the pre-specified burnup steps. In addition, Serpent provides all the

decay constants, and Q-values for the various isotopes. The decay heat can be directly deducted by taking the product of these quantities and summing over all the isotopes.

6.1.1.1 ABR Fuel Assembly

As a first step, a single 3D fuel assembly of the ABR core was modelled in Serpent (Figure 5.3). The fuel assembly was modelled with 6 variations from the nominal operational conditions (fuel temperature of 807 K, coolant density of 2.2272E-02 atoms/barn-cm). The perturbations included high (1200 K) and low (600 K) fuel temperatures, and high (110% and 120%) and low (90% and 80%) values of the original coolant density. The fuel assembly was depleted for one full cycle of 328.5 days at full power, followed by a period of zero power to account for decay heat. The results in the following figures present the corresponding decay heat power:

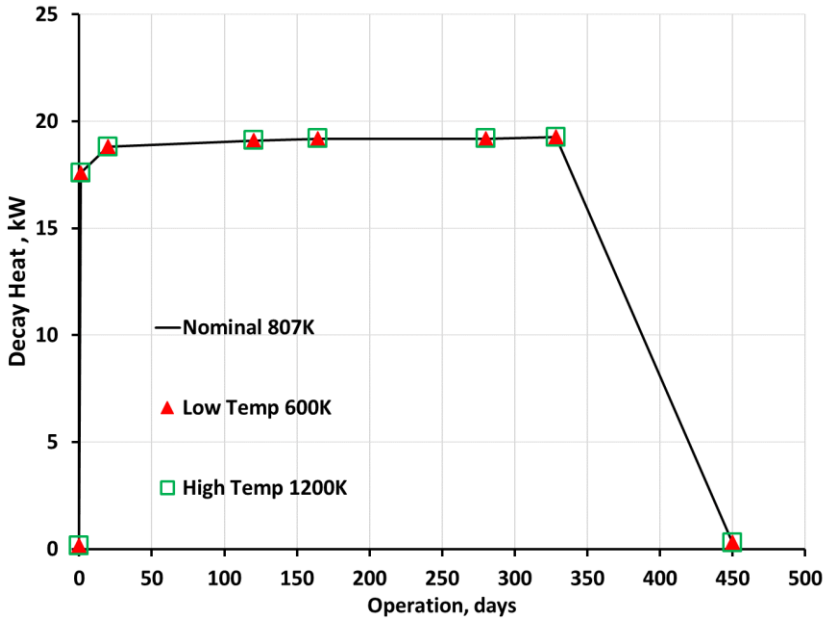


Figure 6.1 Total Decay Heat, Varying Fuel Temperature

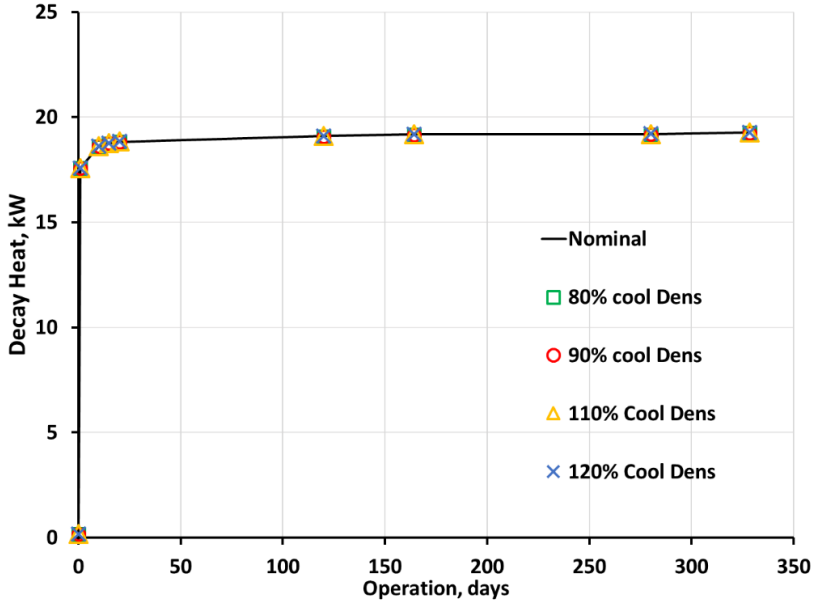


Figure 6.2 Total Decay Heat During Operation, Varying Coolant Density

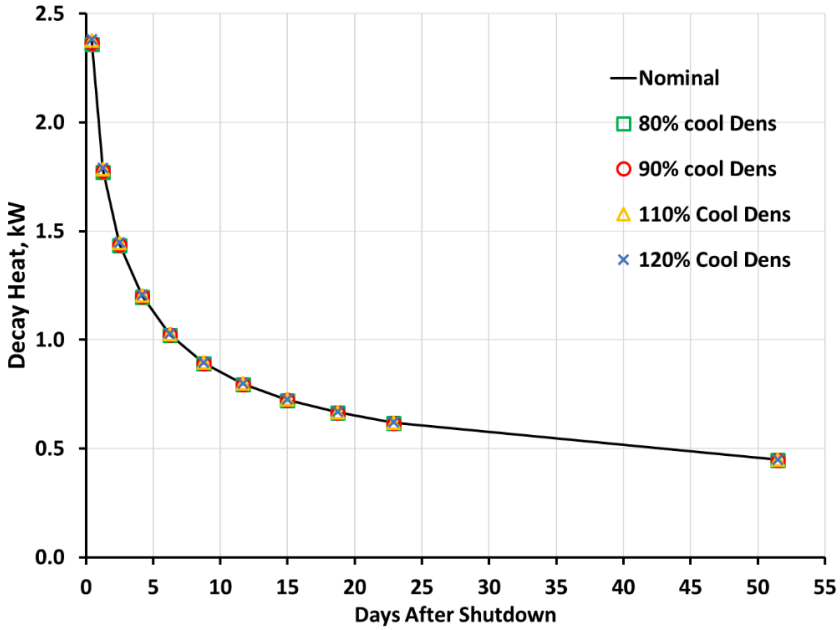


Figure 6.3 Total Decay Heat After Shutdown, Varying Coolant Density

Figure 6.1, Figure 6.2 and Figure 6.3 show no visible variation in the decay heat calculated by Serpent for varying fuel temperatures and coolant densities. To better

visualize the behavior, the difference between varying coolant density cases to the nominal case is plotted for the operational scenario and after shutdown:

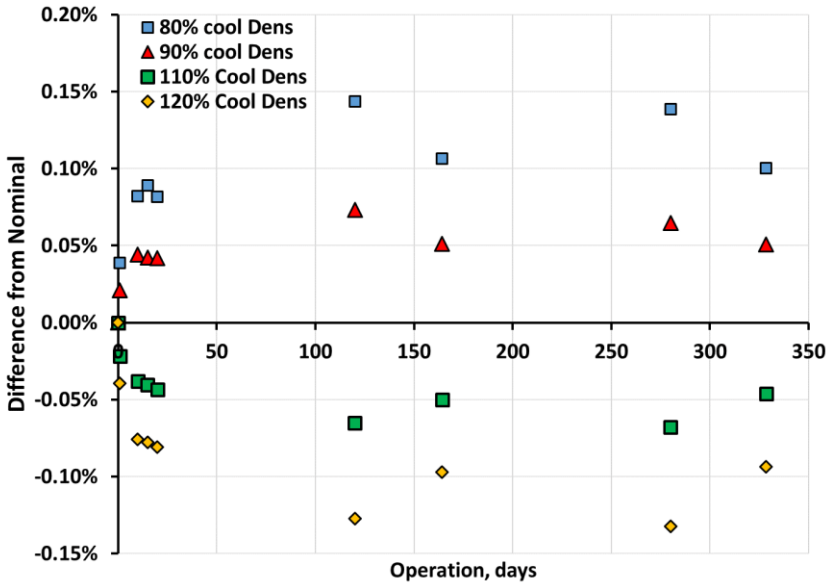


Figure 6.4 Relative Difference in Decay Heat During Operation

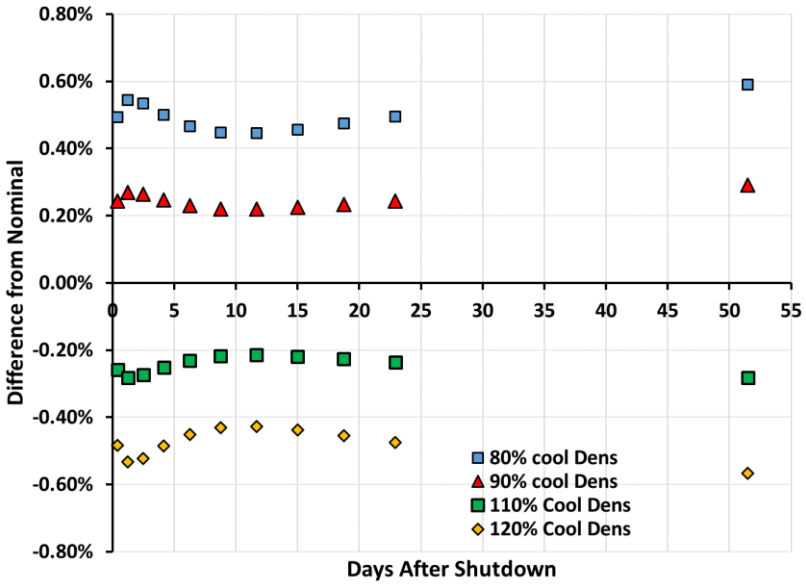


Figure 6.5 Relative Difference in Decay Heat After Shutdown

6.1.1.2 Method Validation by Comparison to PWR Fuel Pin

Despite there being a visible trend with the perturbation applied in the ABR 3D fuel assembly case, the difference is quite small even for large variations in coolant densities. In order to verify that operational conditions indeed have only a small impact in decay heat in fast systems, a standard UO_2 Pressurized Water Reactor (PWR) pin enriched to 3.95%, in which higher variation is expected, is modeled with perturbed fuel temperatures and coolant densities. A more significant discrepancy would indicate that the approach is indeed valid and that fast reactors are less susceptible to change in operational conditions. The fuel pin was depleted for roughly the lifetime of the fuel in a light water reactor, which in this case was stretched out to 1200 days at full power. The nominal operational conditions were taken to have a fuel temperature of 900 K and a water density (coolant and moderator in this case) of 0.7005 g/cm^3. The perturbations applied to the model included reducing the water density to roughly 57% and 86% of the original, and also varying the fuel temperature to a high of 1500 K and a low of 600 K.

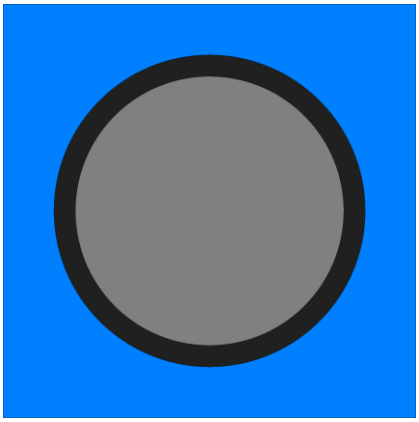


Figure 6.6 PWR Pin Model

For this study, the cumulative energy produced after the point of interest is calculated and plotted. The “set depmtx” card in Serpent was turned on during the depletion of the fuel pin, which produces a detailed tally of the isotopic composition in the fuel at each burnup step. These atomic densities were then taken at two points, after 600 and 1095 days of operations, as to simulate the decay heat at an unexpected transient in the middle of the cycle, as well as at the end of the cycle. These atomic densities were then multiplied by the decay constants, and the Q-values, and the decay heat was calculated as to simulate a transient (followed for 3 days) and fuel after removal from core (followed for 260 days).

A similar analysis was performed for a 2D fuel pin for the sodium fast reactor (SFR). The nominal conditions were taken to be the same as in the previously described 3D fuel assembly (fuel temperature of 807 K, coolant density of 2.2272E-02 atoms/barn-cm). The perturbations applied were 80% and 60% of the original coolant density, and a high fuel temperature of 1500 K and low fuel temperature of 600 K.

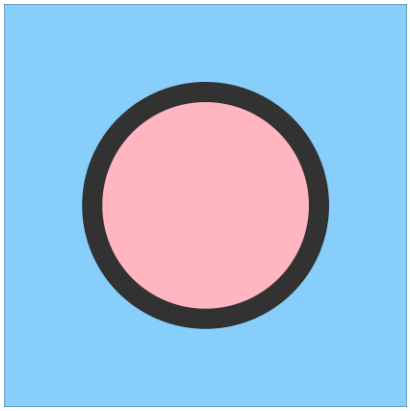


Figure 6.7 ABR Serpent fuel Pin Model

6.1.1.3 Results

As described in Section 6.1.1.2, the differences in the decay heat calculated by Serpent caused by the perturbations were close to negligible. In order to verify this behavior, a PWR fuel pin is modeled. The results presented from here compare the cumulative energy generated after the time of interest. First, Figure 6.8 and Figure 6.9 present the decay heat energy generated once the reactor is shutdown prematurely (e.g. during a LOCA).

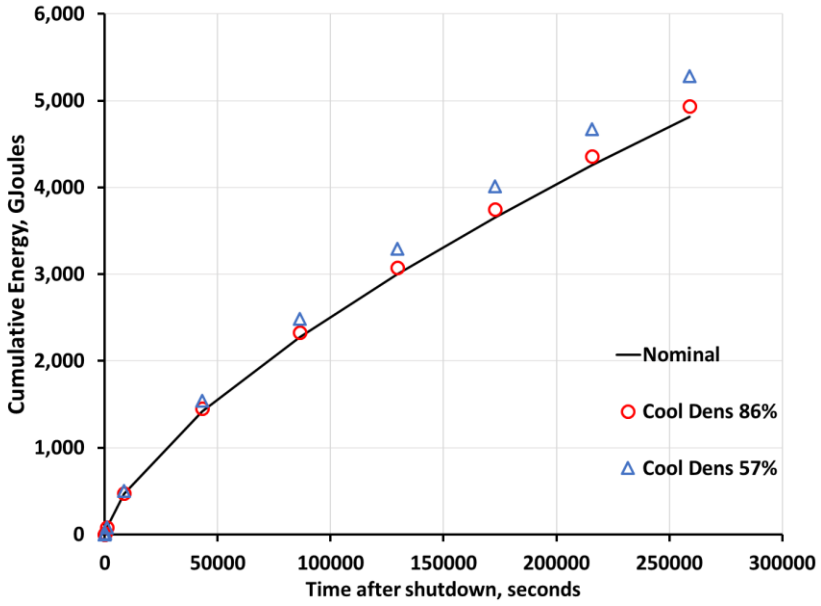


Figure 6.8 Cumulative Energy Production with Varying Water Density, PWR Transient

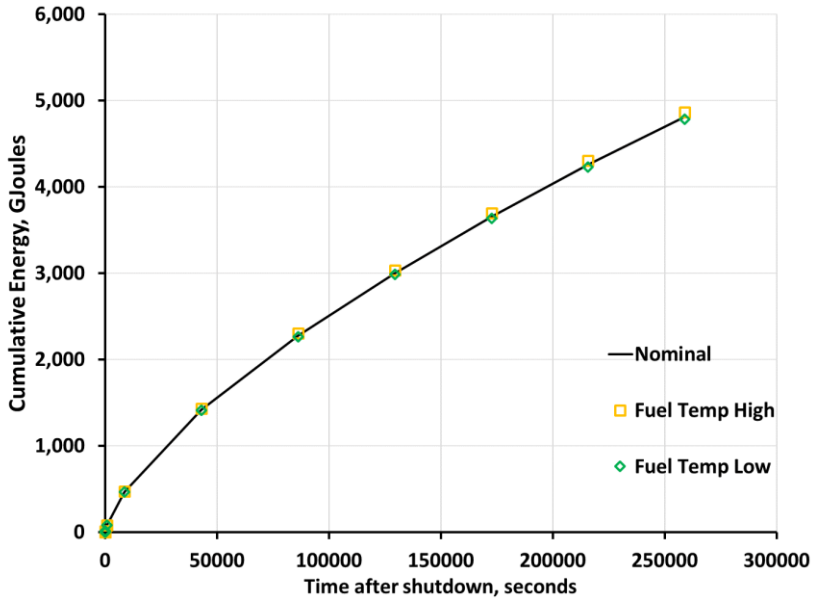


Figure 6.9 Cumulative Energy Production with Varying Fuel Temperature, PWR Transient

Figure 6.10 and Figure 6.11 represent the decay heat energy generated after shutdown at the end of cycle:

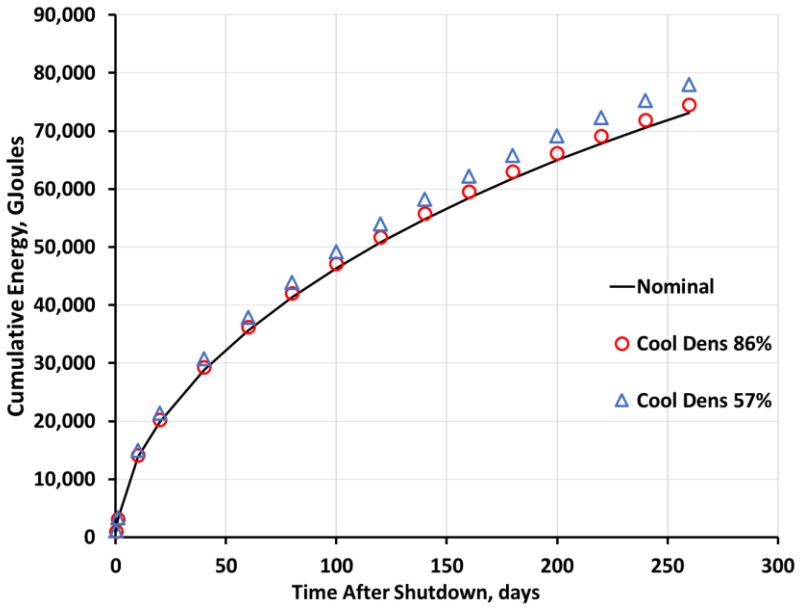


Figure 6.10 Cumulative Energy Production with Varying Water Density, PWR EoC

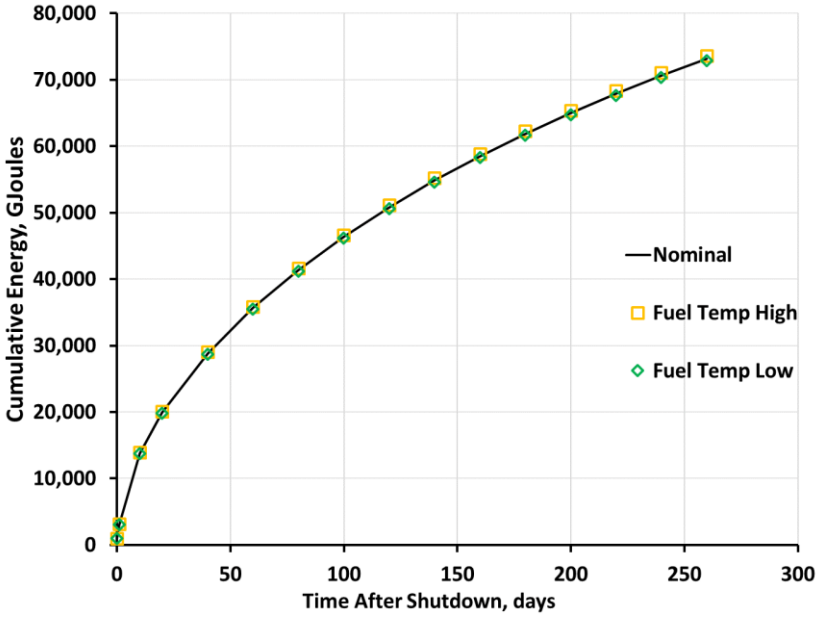


Figure 6.11 Cumulative Energy Production with Varying Fuel Temperature, PWR EoC

It can be seen that the total energy produced remains close to constant when the PWR fuel temperature is perturbed, with an average change of less than 1% for both cases analyzed. However, perturbation in coolant density leads to a more significant variance in energy generated. For the transient scenario, the case with 86% coolant density has a difference in energy produced of 2.51%, while the case with 57% coolant density has a difference of 9.71% at the end of the transient. As for the shutdown scenario, the case with 86% coolant density lead to a difference in cumulative energy produced of 1.95% at 260 days after shutdown, and similarly a difference of 6.63% is seen with the 57% coolant density case.

As previously mentioned, in order to verify that in fast systems the effect of operational conditions has a low impact in decay heat calculations, a similar analysis was performed for the ABR fuel pin, varying coolant density and fuel temperature. The transient scenario was taken from the fuel composition after 300 days of full power

operations, while the EOC case was taken from the fuel composition after 640 days of full power operations. Results for cumulative energy produced are plotted:

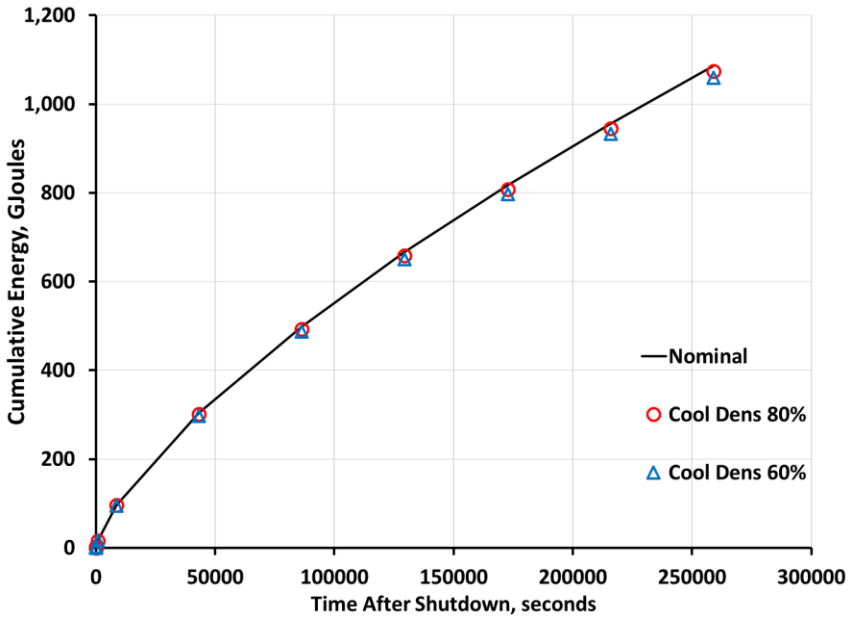


Figure 6.12 Cumulative Energy Production with Varying Sodium Density, ABR Transient

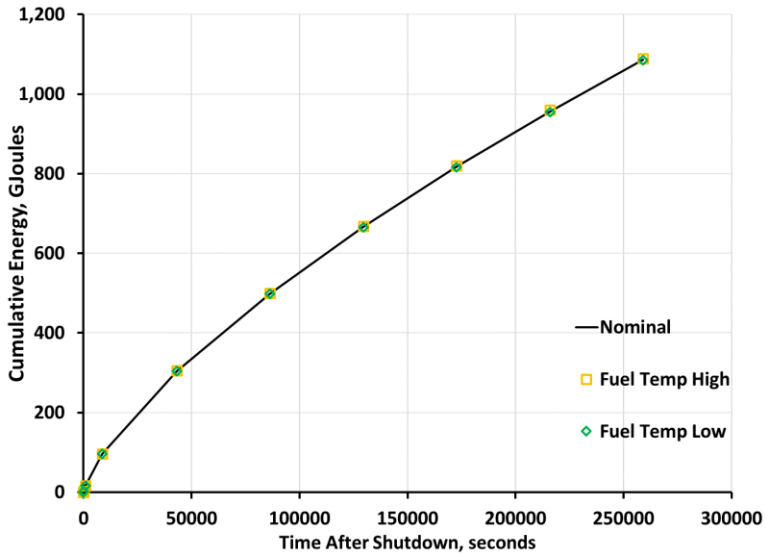


Figure 6.13 Cumulative Energy Production with Varying Fuel Temperature, ABR Transient

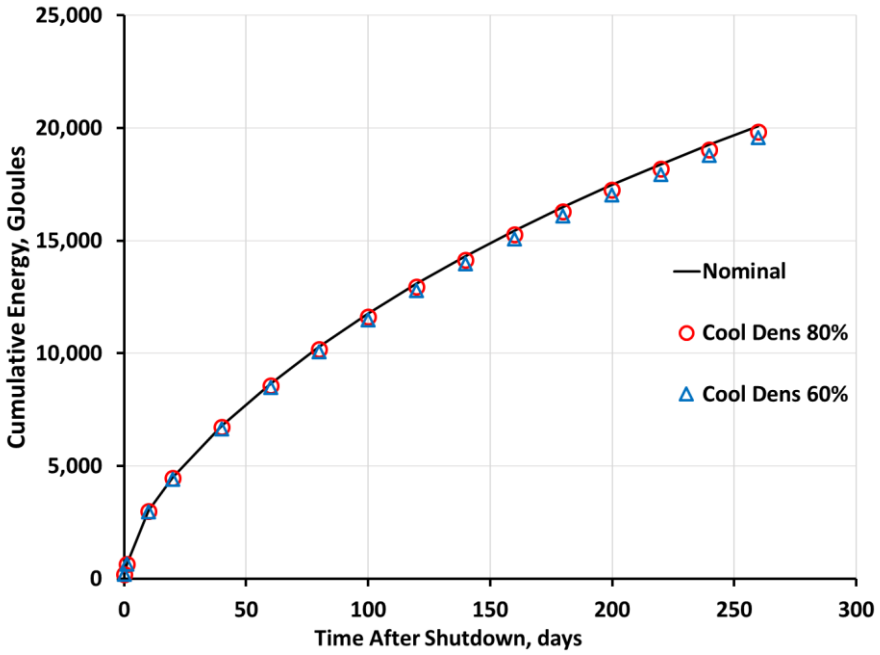


Figure 6.14 Cumulative Energy Production with Varying Sodium Density, ABR EoC

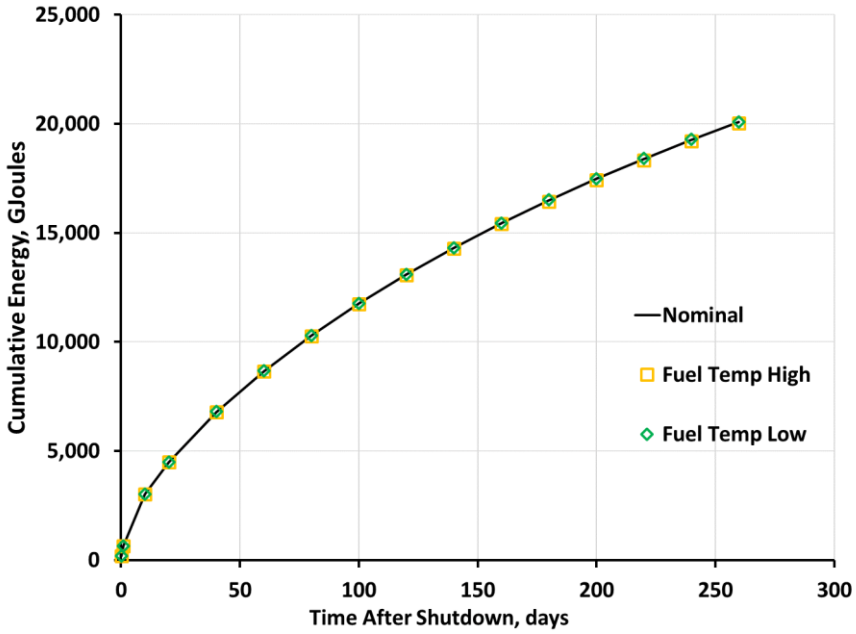


Figure 6.15 Cumulative Energy Production with Varying Fuel Temperature, ABR EoC

As in the case with the PWR pin, perturbing fuel temperature is shown to have negligible impact on cumulative energy production, with an average deviation of less than

0.3% on both cases analyzed. However, unlike the PWR case, the perturbation in coolant density demonstrates a much smaller impact in cumulative energy produced. In the case of the transient scenario, the 80% coolant density case leads to a difference of 1.16% from the nominal case at the end of the transient, while the 60% coolant density case leads to a difference of 2.48%. In the shutdown scenario, the 80% coolant density case leads to a difference of 1.17% in total energy produced after 260 days, and the 60% coolant density case has a 2.51% difference in cumulative energy produced at the same point in time.

Given that even at extreme cases where the coolant density is maintained at 60% and considering that there is some inherent error associated with the Monte Carlo calculation, the decay heat in fast systems are largely unaffected by variations in coolant density and fuel temperatures. Therefore, for the rest of the analysis using the Serpent-DYN3D code sequence, the cases will be carried out for nominal fuel temperature and coolant density. Before performing the decay heat analyses, the microscopic cross sections must be generated.

6.2 Micro-Depletion and Decay Heat Calculation Methodology in DYN3D

The lattice transport - nodal diffusion sequence is a widely used approach for performing full core calculations. Typically, homogenized macroscopic cross-sections are generated by a lattice code and tabulated against the operational parameters (*e.g.*, coolant density, fuel temperature, burnup, etc.). In order to solve the depletion problem, microscopic cross-sections must also be mapped against the operational conditions. DYN3D uses both macro- and microscopic cross-sections sets to perform coupled neutron diffusion-T/H-burnup analysis on a core level.

The coupling scheme implemented in DYN3D relies on the quasi-static approach, in which multiple discrete time-steps are used to perform fuel cycle simulations. At the beginning of each time-step, DYN3D performs coupled neutron diffusion-T/H steady state calculations, with an optional criticality search. The obtained flux and temperatures distributions are considered constant during the time-step. These distributions are used to obtain local burnup and isotopic concentrations at the end-of-step (EoS). Finally, the EoS burnup distribution is used to obtain the updated macro- and microscopic cross-sections and the procedure is repeated for all the subsequent steps.

The microscopic data generation stage is described in Section 6.3, whereas the remainder of this section focuses on describing the general depletion calculations to obtain the nodal nuclide field, $\mathbf{N}(\mathrm{t})$, as a function of time. DYN3D obtains the nodal nuclide vector $\mathbf{N}(\mathrm{t})$, composed of elements $N_{\mathrm{j}}(\mathrm{t})$, representing the atom density of nuclide j, by solving the depletion problem formulated in Equation 6.1:

$$\frac{dN_j(t)}{dt} = -\lambda_j N_j(t) - \bar{\sigma}_{a,j}\phi N_j(t) + \sum_{\substack{k=1 \\ k\neq j}} \left(\lambda_{k\to j} + \bar{\sigma}_{k\to j}\phi\right) N_k(t) \qquad 6.1$$

where, λ_j is the decay constant of nuclide$j$, $\bar{\sigma}_{a,j}$ is the energy average absorption cross-section, ϕ is the total flux, $\lambda_{k\to j}$ is the decay constant from nuclide k to nuclide j, and $\bar{\sigma}_{k\to j}$ is the average transmutation cross-section of the nuclide k that leads to the production of nuclide j. The set of multiple first order differential equations, Equation 6.1, can be represented in a matrix form:

$$\frac{d\mathbf{N}(t)}{dt} = \mathbf{A}\,\mathbf{N}(t) \tag{6.2}$$

where $\mathbf{A}$ is the transmutation matrix, which consists of the radioactive decay and neutron induced reactions rates terms. The solution of Equation 6.2 is obtained using the matrix exponential, as described in Equation 6.3, assuming constant reaction rate during time-step Δt.

$$\mathbf{N}(t) = \exp[\mathbf{A}\Delta t]\,\mathbf{N}(t_0) \tag{6.3}$$

where, $\mathbf{N}(t_0)$ describes the known nuclide vector at time t_0, and Δt is the length of the depletion time-step. Detailed discussion on the matrix components and their generation is given in Section 6.3.1.

The method implemented in DYN3D to solve the matrix exponential (*i.e.*, $\exp[\mathbf{A}\Delta t]$) is Chebyshev rational approximation [18], [19]. This method was proven to be particularly suitable [20] for fuel depletion and radioactive decay problems.

The macroscopic cross-sections for nodal diffusion calculation, as well as microscopic cross-sections for depletion calculation, are interpolated from a pre-generated library using local burnup and operational parameters. The operational history is taken into account by correcting the macroscopic cross-sections using local nuclide concentrations [21]. Detailed spatial nuclide content distribution also allows to calculate the radioactive decay heat via the isotopic summation method, without relying on semi-empirical correlations, as described in Equation 6.4:

$$P^n(t) = \sum_j N_j^n(t)\lambda_j q_j \qquad 6.4$$

where $P^n(t)$ is the decay heat rate in node n at time t, $N_j^n(t)$ is the time dependent concentration of nuclide j in node n, and q_j is the energy release per decay of nuclide j. To capture the contribution of all the heat producing nuclides, DYN3D explicitly tracks over 1,200 isotopes (by default) via their decay or transmutation reactions. However, the transmutation chains can be easily extended, provided that the corresponding homogenized microscopic cross-sections are supplied to DYN3D by the lattice code.

This method was implemented in DYN3D and tested only on simple 2D infinite lattices [22]. The main goal of this work is to demonstrate the applicability of this method to predict spatial isotopic concentrations and decay heat distributions in 3D full core case.

6.3 Microscopic Cross Section Generation

The macroscopic cross-section generation methodology was introduced and described in Chapter 5.2. The same approach is used for the required macroscopic cross-sections used in this portion of the analysis. The remainder of this section focuses on the generation of the microscopic cross-sections used for the depletion and decay heat analysis.

6.3.1 Generation of the Transmutation Matrix

This section describes the data preparation associated with solving the depletion problem (*i.e.*, Bateman equations). More specifically, this section describes the procedure adopted to assemble the transmutation matrix and the corresponding simplifications made

to ease the data management. In this study, a total of ~1,400 nuclides are included in the transmutation matrix, which is a sparse matrix (denoted as **A**) as shown in Figure 6.16.

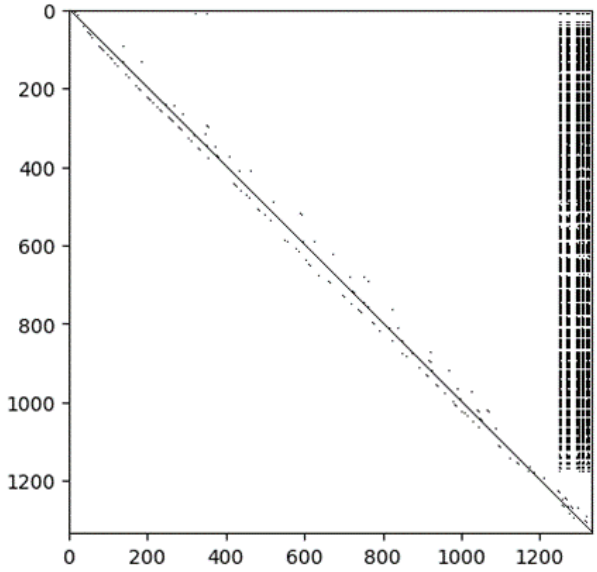


Figure 6.16 Sparsity of the Transmutation Matrix.

Figure 6.16 shows a dense diagonal ($A_{j,j}$ elements) that represents the decay constants (λ_j) for each isotope j and the group absorption reaction rates as shown in Equation 6.5.

$$A_{j,j} = -\left(\lambda_j + \sum_g \sigma_{a,j}^g \phi_g\right) \tag{6.5}$$

where, ϕ_g is the neutron flux in group g and the summation is performed over all the energy groups (one-group absorption reaction rates). The off-diagonal, $A_{k,j}$, elements are composed by including the decay from isotope k to j ($\lambda_{k\to j}$) and the transmutation rate ($\sigma_{k\to j}^g \phi_g$) from isotope k to j.

$$A_{j,k} = \sum_k (\lambda_{k\to j} + \sum_g \sigma_{k\to j}^g \phi_g)\ ,\ \ k \neq j \tag{6.6}$$

Lastly, the dense array of columns in Figure 6.16 correspond to the production of fission fragments from fissions events originated from different fissionable elements as described by Equation 6.7:

$$A_{j,k} = \sum_{g} \gamma_g^{k \to j} \sigma_{f,k}^{g} \phi_g \ , \quad k \neq j \tag{6.7}$$

where, $\gamma_g^{k \to j}$ is the energy dependent fission yield that leads to the production of isotope j (*e.g.*, Xe-135), given that a fission event occurred in isotope k (*e.g.*, Pu-239).

Some key points related to the compilation of the transmutation matrix are given below:

6.3.1.1 Transmutation Matrix

The transmutation matrix, which includes around 1,400 isotopes, was obtained directly from Serpent. The latter does not use a fixed number of nuclides or decay and depletion chains to construct a transmutation matrix. Instead, Serpent builds the transmutation chains automatically starting from the initial fuel composition and using the reaction modes described in radioactive decay and cross section data libraries [13]. Under the current implementation, the decay matrix in DYN3D is fixed based on the initial isotopic composition of the ABR fuel. There is no error associated with the mentioned simplification. However, if a completely different fuel composition is taken (*e.g.*, thorium-based fuel), the transmutation matrix should be re-generated.

6.3.1.2 Microscopic Cross-sections

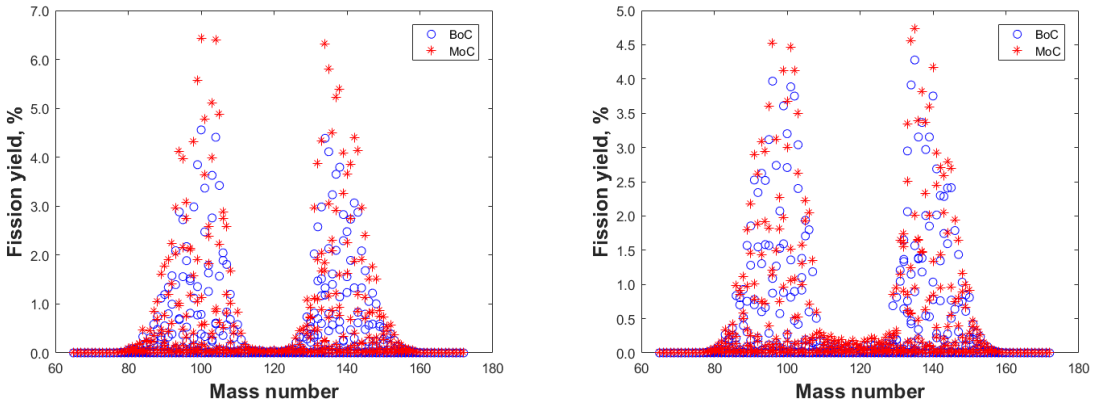
Serpent allows for the generation of homogenized microscopic cross-sections for any isotope and reaction ("mdep" card) without the need to define dedicated detectors/tallies. This option was exploited to generate tabulated few-group microscopic cross-sections with the same energy structure used for the macroscopic quantities. Only the (n,γ), (n,fission), (n,2n), (n,3n), and (n,α) reactions were considered in the transmutation chains. Microscopic cross-sections were generated for precisely 290 of the 1,400 isotopes corresponding to the isotopes included within the ENDF/B-VII cross section library. It must be emphasized that Serpent includes many more reactions, such as (n,t), which were not considered in DYN3D. However, the error associated with this simplification has a negligible effect on isotopic concentration of most nuclides, with higher impact on the lighter nuclides which are more likely to undergo some of these rare reactions.

6.3.1.3 Fission Yields

During calculations, Serpent obtains energy-dependent fission yields directly from the ENDF formatted fission yield data files. Typically, fission yield data is specified for several particular incident neutron energies. To obtain the effective fission yields for depletion calculations, Serpent interpolates the energy-dependent fission yield for every fissionable isotope that undergoes a fission event during the transport tracking routine. As a result, the fission yields in Serpent vary with burnup and spectrum and are composition dependent. This continuous energy approach cannot be reproduced exactly in DYN3D nor in any diffusion code because the code relies on energy discretization. Consequently, it was decided to generate energy-independent sets of fission yields for every fissionable isotope

based on the Serpent beginning-of-life data. Finally, Serpent produces fission yield values at every burnup point, leading to prohibitively large sets. Therefore, it was also decided to keep the fission yields constant throughout the depletion cycle.

This simplification in fission yields treatment leads to discrepancies in the fission products concentrations. Figure 6.17 presents the fission yields distribution for Pu-239 and U-238 at different burnup points. It is clear that this difference is propagated through the depletion analysis and has non-negligible impact on some specific fission products. Potential approaches to avoid storing the fission yields as a function of operational conditions were not investigated in this work, but various solutions can be explored in future campaigns.



(a) Pu-239 (b) U-238

Figure 6.17 Weighted Fission Yields at Beginning-of-Cycle (BoC) and Middle-of-Cycle (MoC)

6.3.2 Data Processing Requirements

The data storage and processing requirements associated with tabulating macroscopic quantities are fairly negligible. However, this is not the case if microscopic

quantities are added to the tabulation process. Energy dependent microscopic cross-sections of different type (*e.g.* σ_f) must be stored for hundreds of nuclides and mapped against the operational conditions. For each fuel type, a unique library with microscopic parameters must be generated. In addition, fission yields for each possible parent-daughter pair must also be included in the pre-generated libraries.

In the current analyses, 10 unique fuel types and hence libraries were considered. The cross-sections in each library were generated for 290 isotopes, five reaction types, and 24 energy groups, resulting in a large amount of data.

It must be pointed out that Serpent generates output files in MATLAB format. The few-group macroscopic cross-sections are printed in the main result file (i.e., "_res.m"), which occupied more than 70 MB in the current study. The microscopic cross-sections are printed in a separate "_mdx.m" file (6 MB) for each depletion step. The results were processed using the serpent-tools package, developed by the CORE group at Georgia Tech, previously described in Chapter 2.4.3.

6.4 Results of Multi Geometry Decay Heat Analysis

The results presented in the following sub-sections were produced for three cases, with increasing levels of complexity. First, an infinite fuel assembly (Section 6.4.1) was investigated due to its simplicity and the ability to accurately predict the few-group fluxes. The next test case (Section 6.4.2) focused on an infinite heterogeneous 3D fuel assembly, in which the spatial flux variation affects the axial decay heat predictions. Finally, the core level results are summarized in Section 0, in which a variable power scheme was applied. It must be emphasized that macro and microscopic cross-sections were tabulated only as a

function of burnup (as opposed to also tabulating the group constants as functions of operational conditions, which was a decision made based on the results of the analysis presented in Section 6.1.1).

6.4.1 2D Lattice

As a first step, an infinite 2D assembly is depleted in Serpent, with cross-sections being generated throughout the depletion cycle. The macro and microscopic cross-sections are then used in DYN3D to repeat the depletion calculations. For this scenario, the power of the simulated system was varied (Figure 6.18) throughout the depletion to investigate the effect this would cause on using the code sequence to calculate decay heat. Criticality reported by Serpent and DYN3D is compared and plotted in Figure 6.19. As expected, the agreement is within statistical uncertainty with the difference in reactivity that clearly experiences a stochastic behavior as shown in Figure 6.19.

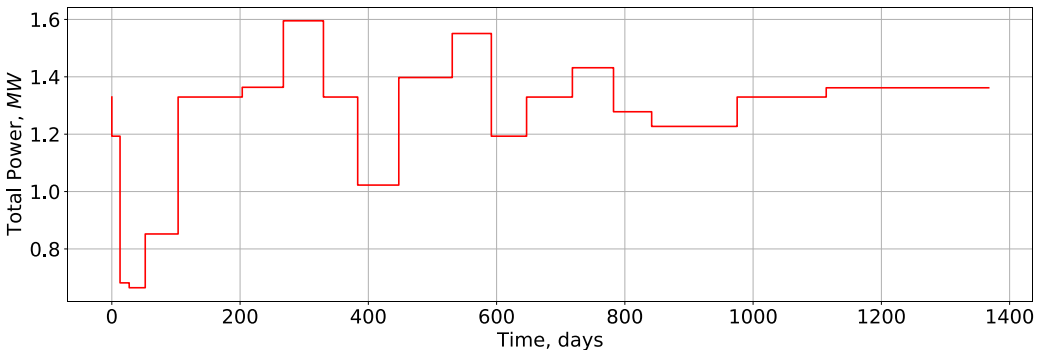


Figure 6.18 Assembly Power Profile Throughout the Cycle

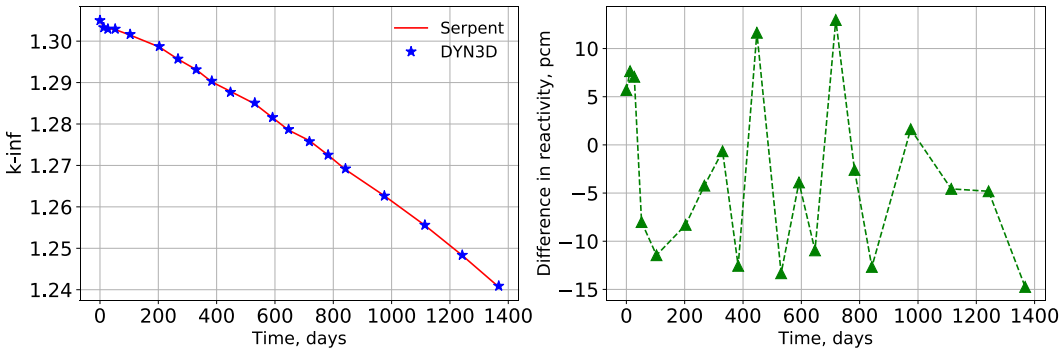
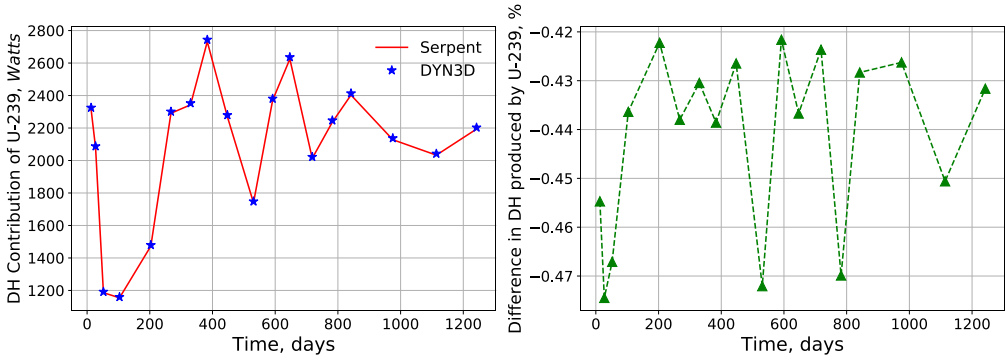
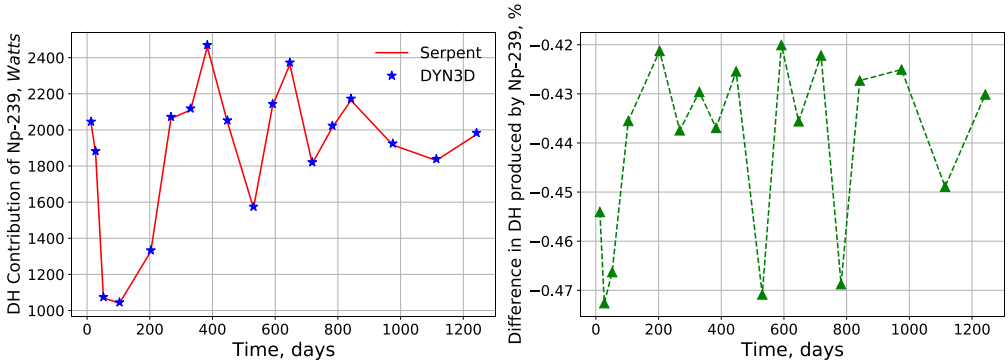


Figure 6.19 Criticality Throughout Depletion Cycle for Serpent vs. DYN3D Infinite Lattice

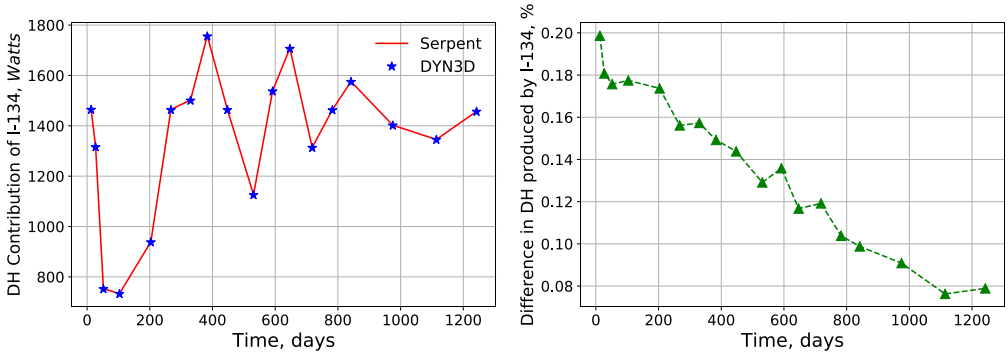
The ability of each code to keep track of isotopic concentration throughout the depletion cycle is crucial for calculation of decay heat. The individual contribution of the top five isotopes to decay heat is compared between Serpent and DYN3D in Figure 6.20. To determine which isotopes constitute the largest decay heat (DH) share, the contribution of each isotope is averaged over the depletion cycle. It can be seen that these isotopes are in very good agreement, with a difference in decay heat produced on the order of 0.3%. It must be pointed out that the isotopes selected for comparison here are only important at the start phase of an accident due to their very short half-life. The list of the most contributing isotopes to decay heat will vary with the time scale of interest, *e.g.*, hours for accident analysis or months to thousands of years for waste storage calculations.



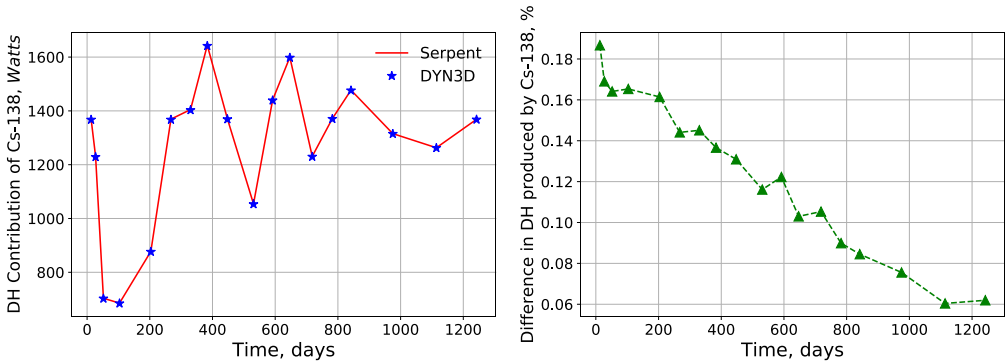
(a) U-239



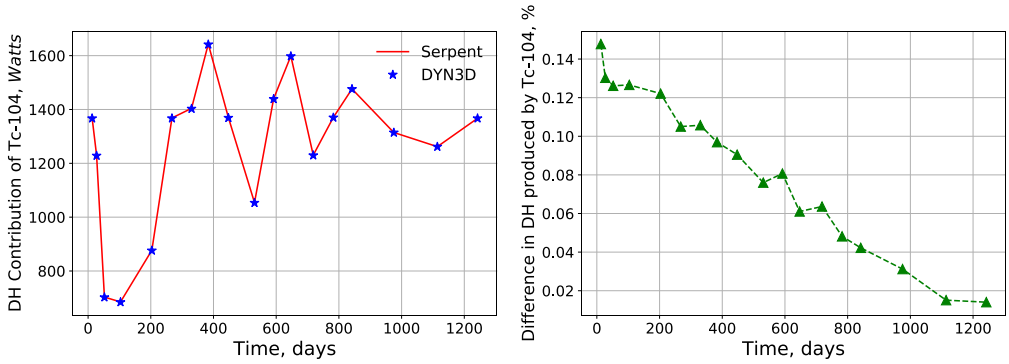
(b) Np-239



(c) I-134



(d) Cs-138



(e) Tc-104

Figure 6.20 Decay Heat Contribution and Difference in Decay Heat Calculated by Serpent and DYN3D for the Five Most Contributing Isotopes

Finally, the total decay heat in the system calculated by Serpent and DYN3D is compared. Very good agreement is achieved, with the difference between codes remaining close to constant throughout the cycle, even though the power is varied greatly in the cycle.

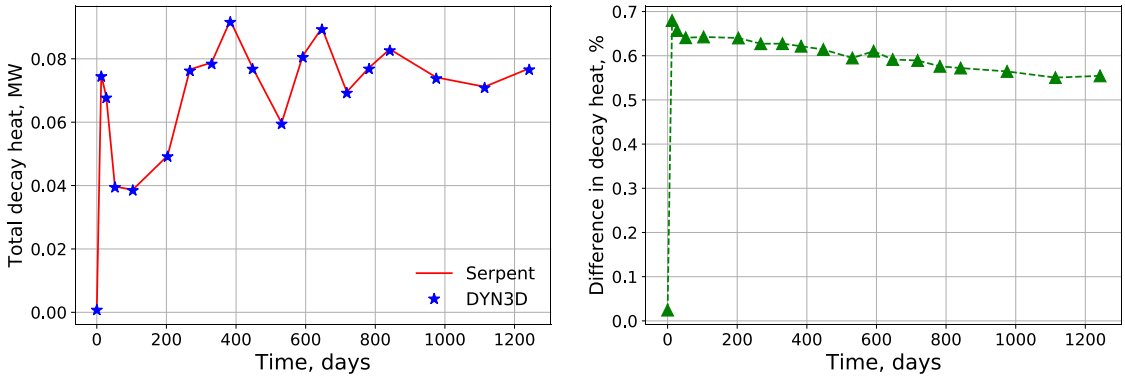


Figure 6.21 Total Decay Heat in the Assembly Throughout Cycle and Difference Between Serpent and DYN3D

6.4.2 3D Assembly

The assembly active fuel region is axially divided into five heterogeneous layers with unique isotopic concentrations, as shown in Figure 5.3. The cross-sections for the fuel are

generated throughout the depletion cycle, while a single zero-burnup cross-sections set is used for non-multiplying regions. The criticality calculated by Serpent and DYN3D are compared in Figure 6.22.

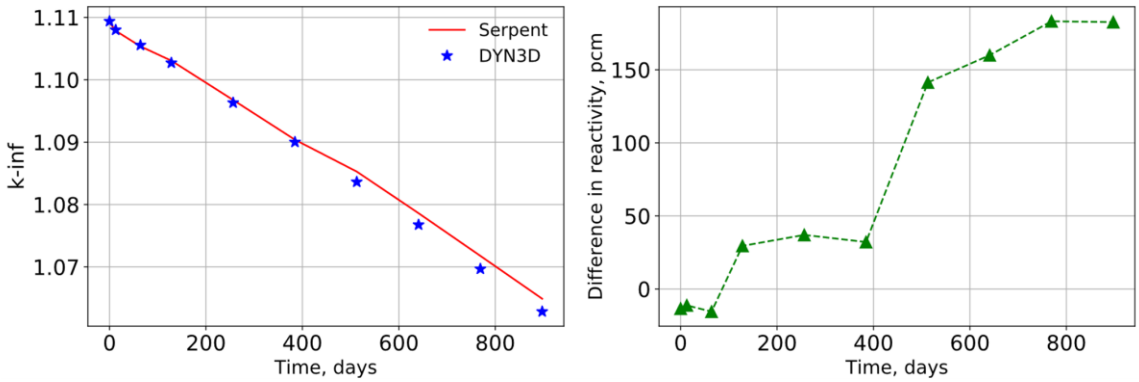


Figure 6.22 Criticality Throughout Depletion Cycle for Serpent vs DYN3D 3D Fuel Assembly

In general, the agreement between the codes is very good, although there appears to be a consistent error build-up as the assembly continues to be depleted. The origin of the discrepancy arises from the fuel-reflector heterogeneity, but this is still considered to be a relatively small difference. For sake of comparison, the average standard deviation reported by benchmark participants at the End of Cycle (EoC) is in the order of 700 pcm [2], therefore certain variation is to be expected when different methods are used. In order to identify the axial layer (fuel region) with the largest contribution to the error, the burnup calculated by each code is compared for each fuel region and is depicted in Figure 6.23. There is good agreement for the burnup calculated by Serpent and by DYN3D in all fuel regions, with a maximum difference of less than 1% throughout the depletion cycle.

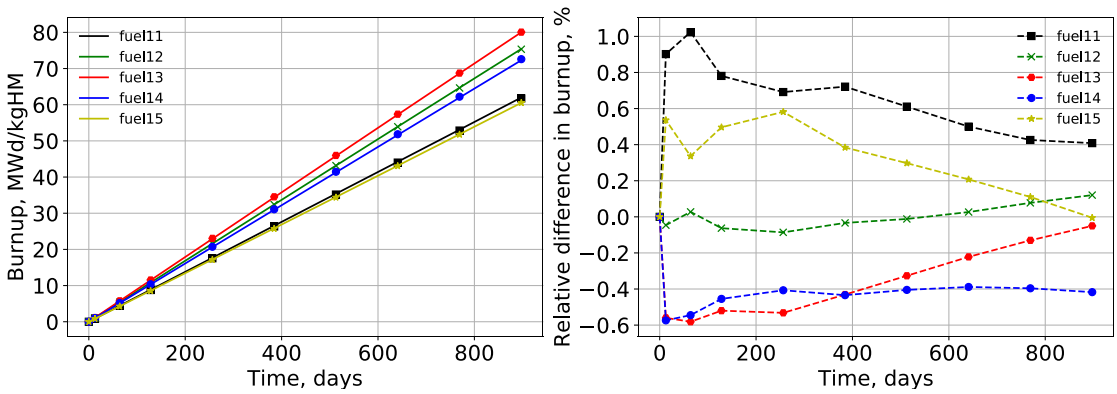


Figure 6.23 Burnup Comparison for Each Fuel Region Throughout Depletion

The isotopic composition calculated by each code is compared by taking the root mean square (RMS) error for each isotope i, over the five layers ($J = 5$) and the nine time-steps ($K = 9$) according to Equation 6.8. The RMS differences for fission products and actinides are shown in Figure 6.24. It should be noted that only isotopes with concentration values above $10^{13} at./cm^3$ were included in the comparison. Very good agreement is found between the isotopic concentrations reported by Serpent and DYN3D, both for fission products and actinides, with a maximum RMS on the order of 0.5% in a few outlier cases.

$$RMS_i = \sum_k \sum_j \sqrt{\frac{100\%}{K \times J} \left[\frac{N_j^{Serpent}(t_k) - N_j^{DYN3D}(t_k)}{N_j^{Serpent}(t_k)} \right]^2} \qquad 6.8$$

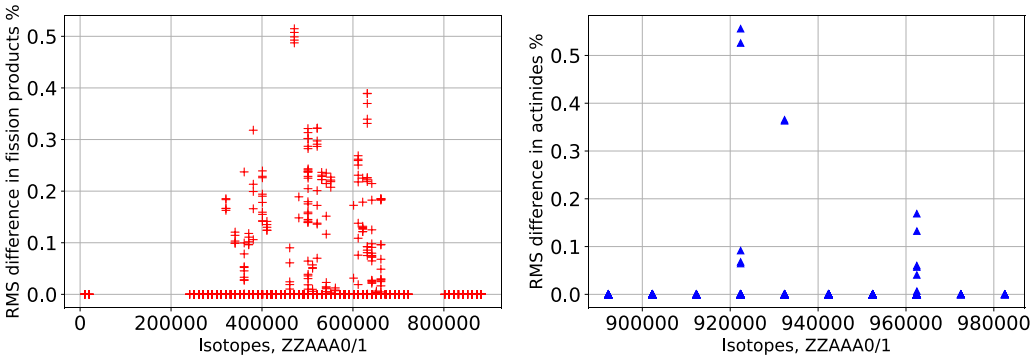


Figure 6.24 RMS Difference in Atomic Density Calculated by Serpent and DYN3D for All Isotopes

Finally, the decay heat generated in each fuel region is calculated by Serpent and by DYN3D, and results are compared in Figure 6.25. A good agreement is observed between Serpent and DYN3D, with the highest difference in decay heat happening at the lower fuel region, at a maximum of 1.6%, but the difference decreases to 0.6% at the end of the depletion cycle. This relatively high discrepancy in decay heat occurs due to the slightly different spatial flux values predicted by DYN3D, which leads to slightly different nuclide densities.

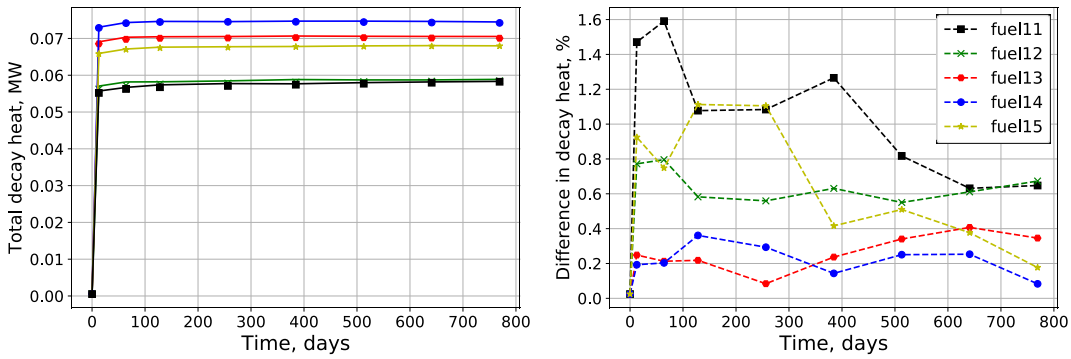


Figure 6.25 Total Decay Heat Generated in Each Fuel Region Throughout the Depletion Cycle

6.4.3 3D Full Core

The reference solution for the full core depletion case is generated with the Monte Carlo code Serpent. The calculations here were performed with fixed temperature distribution (no T/H feedback). For the cross-sections library generation, the fuel assemblies of the inner and outer core (shown in Figure 5.3) were independently depleted for a full cycle, with the cross-sections for each axial fuel region being generated for each burnup step. The cross-sections for non-multiplying regions were assumed to remain constant throughout the cycle. Here, the *B1* leakage corrected cross-sections [9] were applied, only for the fuel assemblies, to account for the neutron leakage and nearly critical core configuration. The arbitrarily chosen operational power history is shown in Figure 6.26.

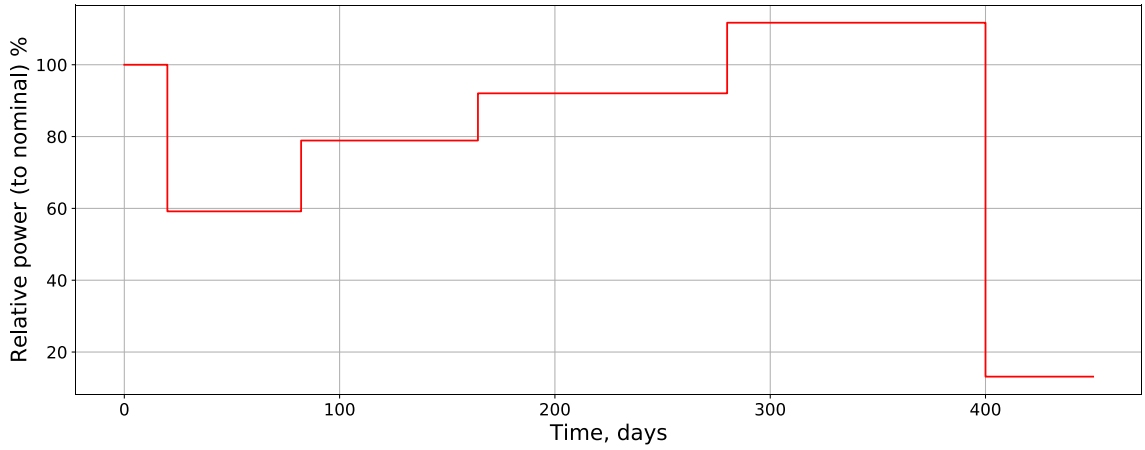


Figure 6.26 Power Level Throughout Depletion Cycle

A good agreement is observed for the criticality values (Figure 6.27) calculated by both codes. The difference in radial power distribution reported by both codes is compared in Figure 6.28 and Figure 6.29 for one quarter of the core. These figures show good

agreement in the radial power results produced by Serpent and DYN3D, with a maximum difference of 1.38% at the Beginning of Cycle (BoC) and 2.11% at the End of Cycle (EoC). The calculation of difference in power distribution was repeated for selected intermediate steps throughout the cycle. The difference in criticality and the radial power RMS error are summarized in Table 6.1.

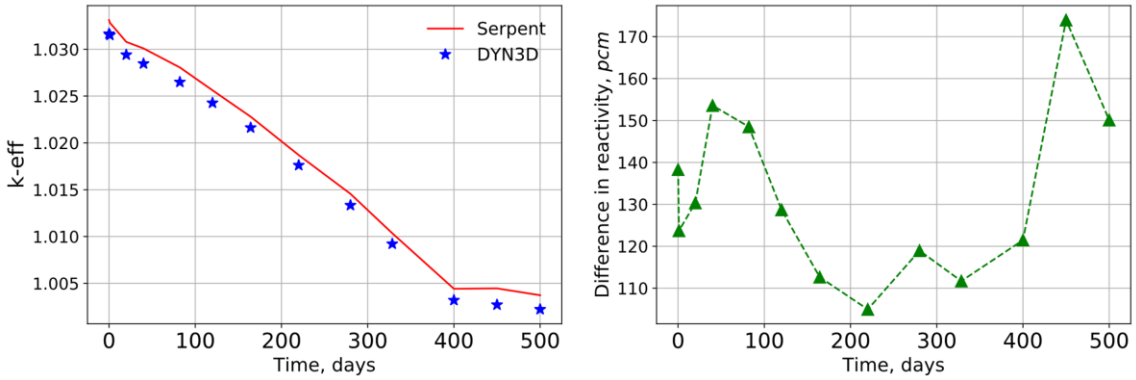
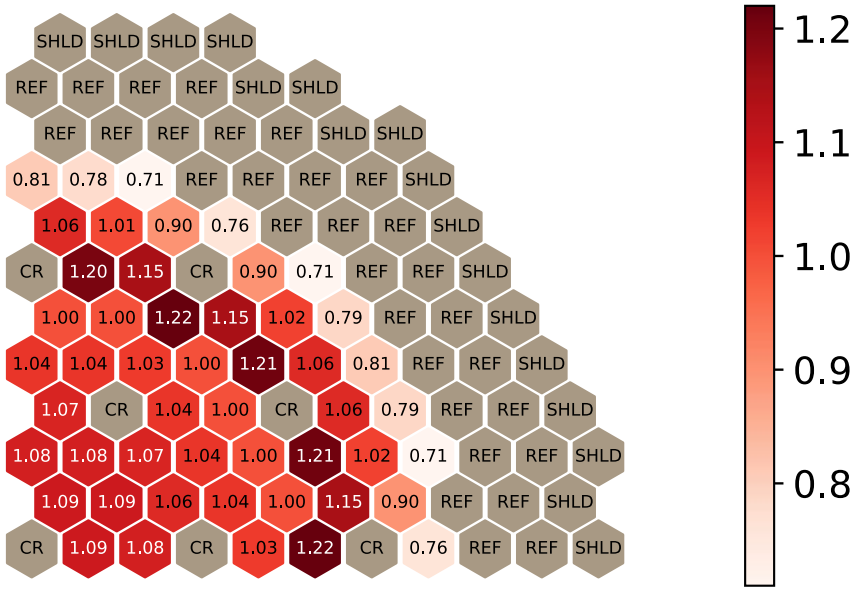
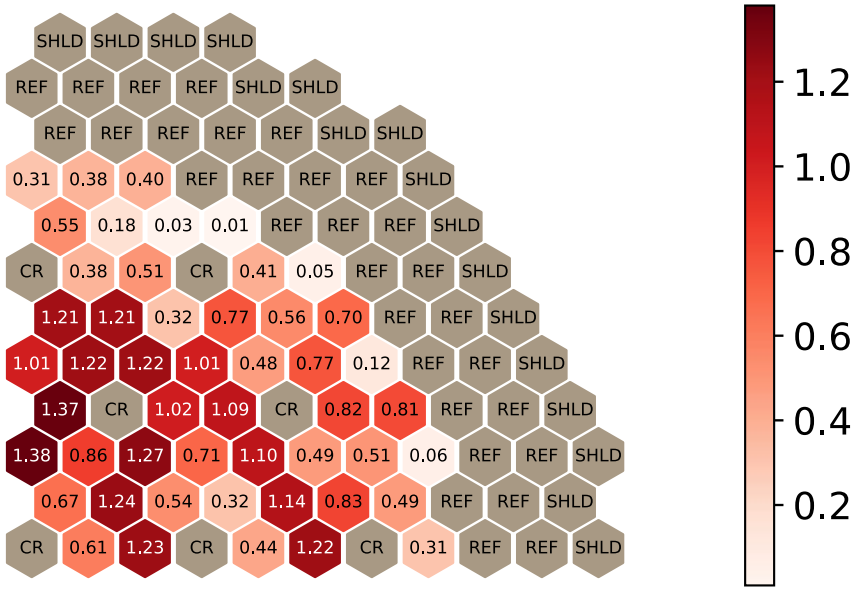


Figure 6.27 Criticality Throughout Depletion Cycle for Serpent vs DYN3D Full Core Model

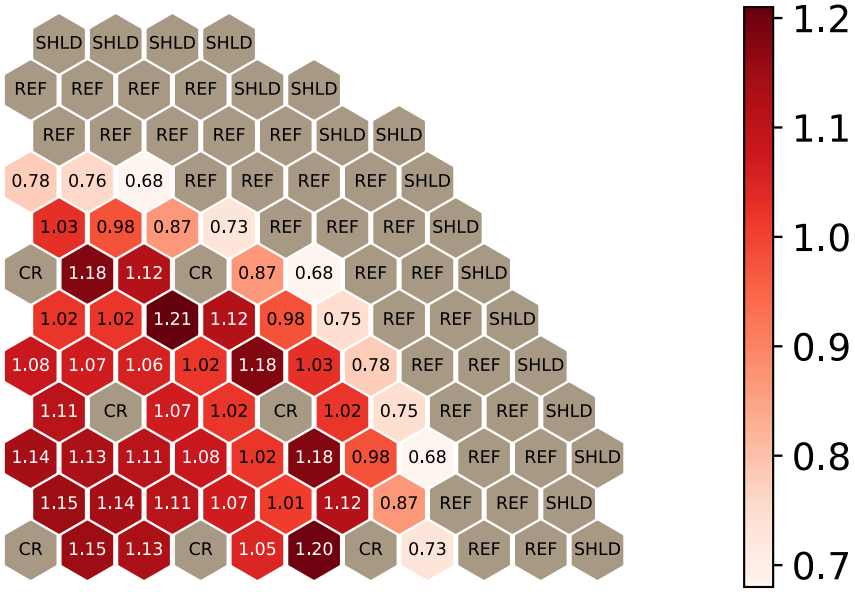


(a) Power Peaking Values

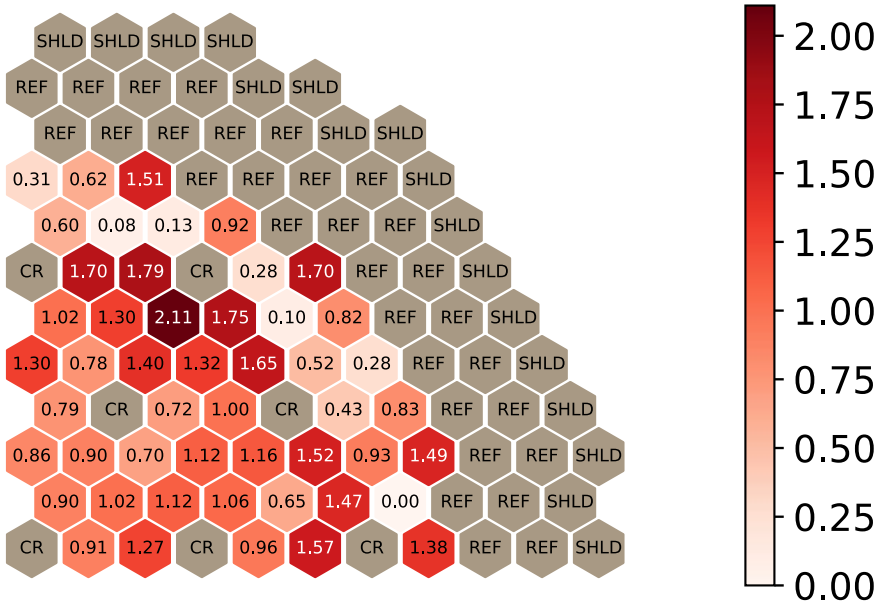


(b) Relative Difference in Power (%)

Figure 6.28 Difference in Radial Power Distribution at Beginning of Cycle



(a) Power Peaking Value



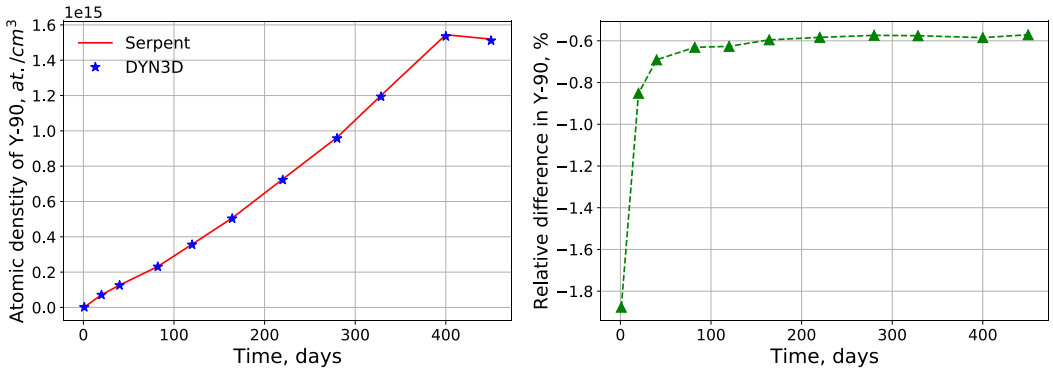
(b) Relative Difference in Power (%)

Figure 6.29 Difference in Radial Power Distribution Towards the End of Cycle (328 days)

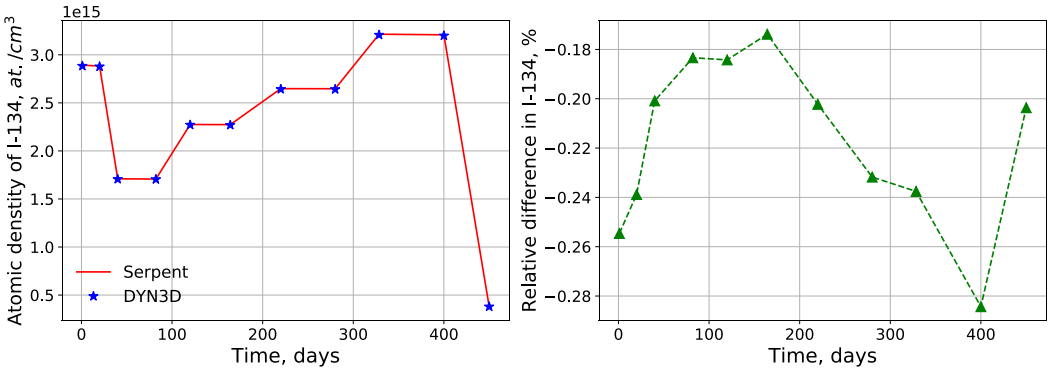
Table 6.1 Serpent vs. DYN3D Comparison for Selected Intermediate Depletion Points

BU, MWd/kg	Time, Days	k_{eff} DYN3D	k_{eff} Serpent	Uncertainty, pcm	Difference, pcm	Radial Power RMS error, %
0	0	1.031646	1.03312	5	138	0.86
3.042	40	1.028463	1.03009	5	154	0.83
6.236	82	1.026493	1.02806	5	148	0.88
12.168	160	1.021623	1.02280	4	113	0.97
24.945	328	1.009211	1.01035	5	112	1.17

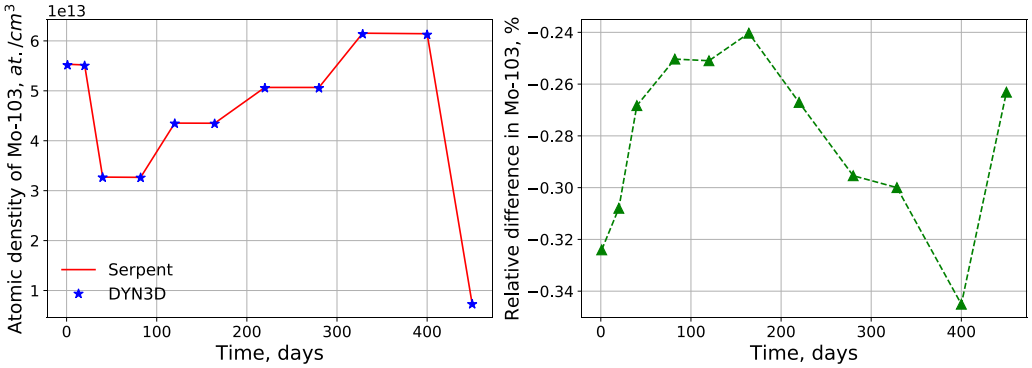
The core average isotopic composition of specific important fission products and actinides are presented in Figure 6.30 and Figure 6.31 respectively.



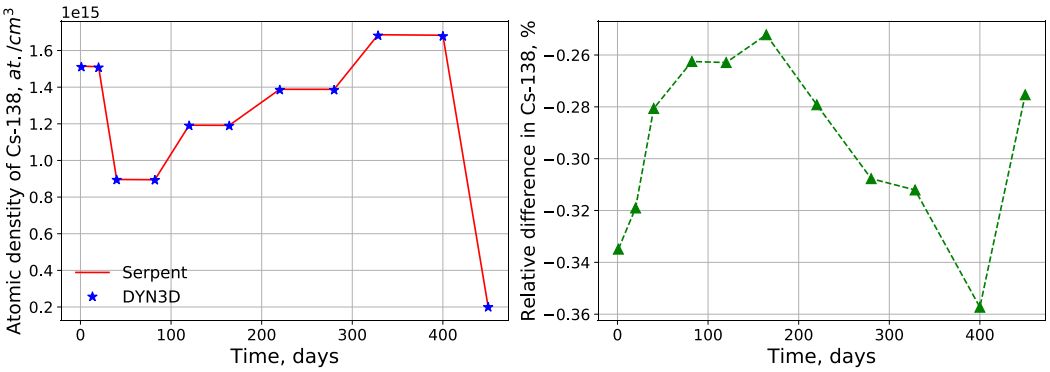
(a) Y-90



(b) I-134

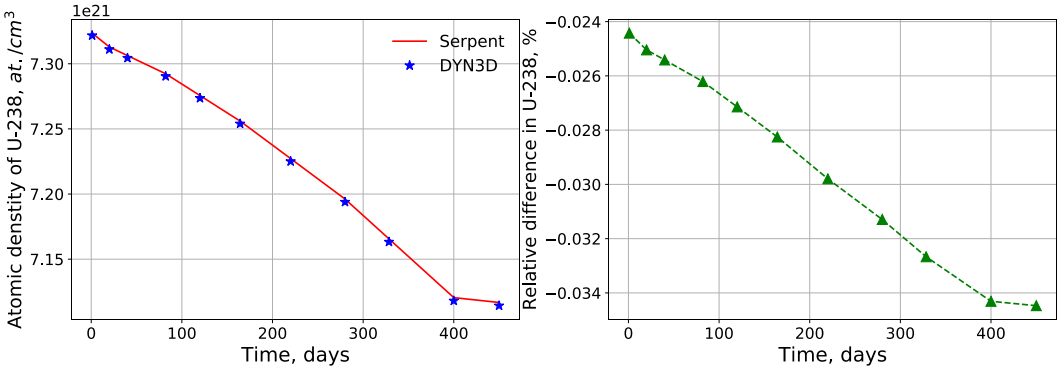


(c) Mo-103

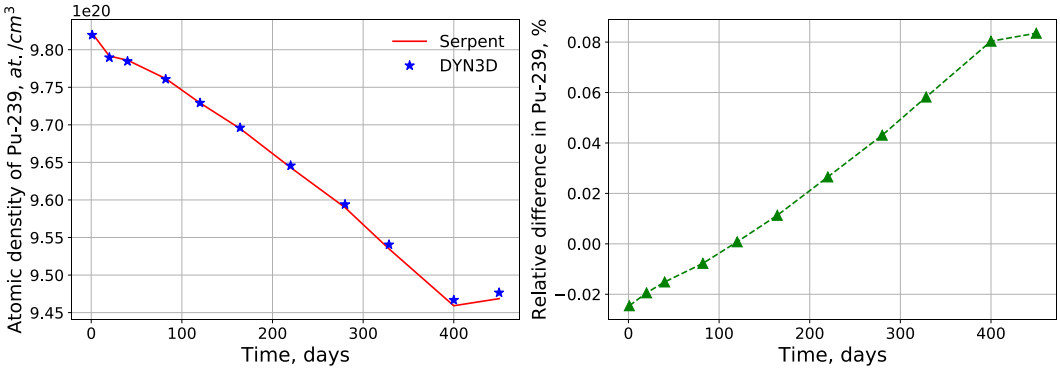


(d) Cs-138

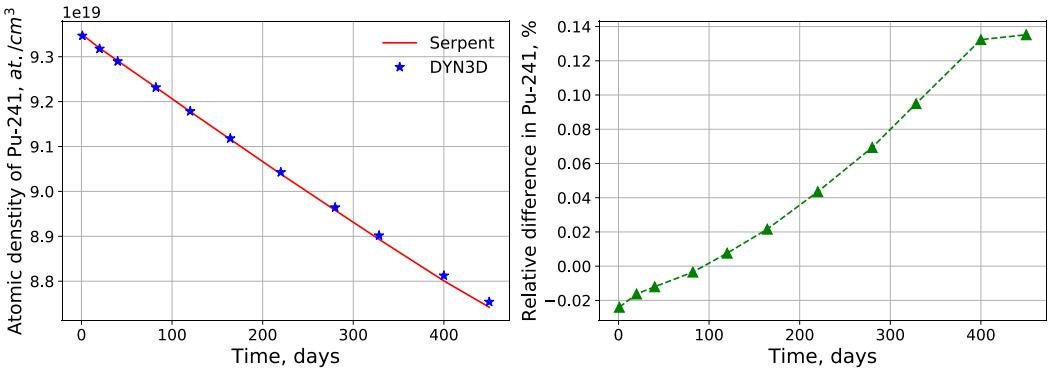
Figure 6.30 Atomic Concentrations and Differences Calculated by Serpent and DYN3D for Various Fission Products



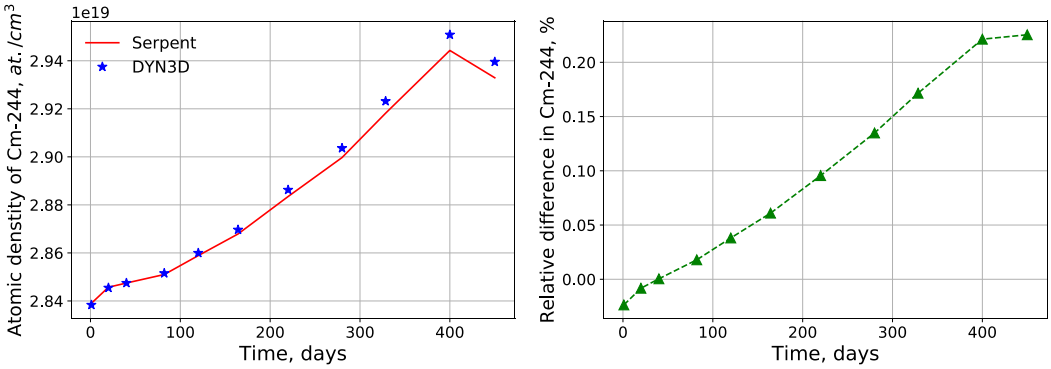
(a) U-238



(b) Pu-239



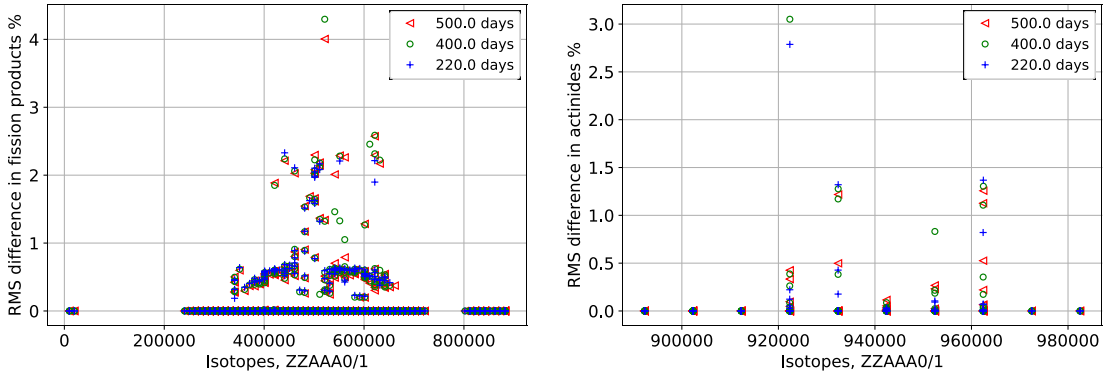
(c) Pu-241



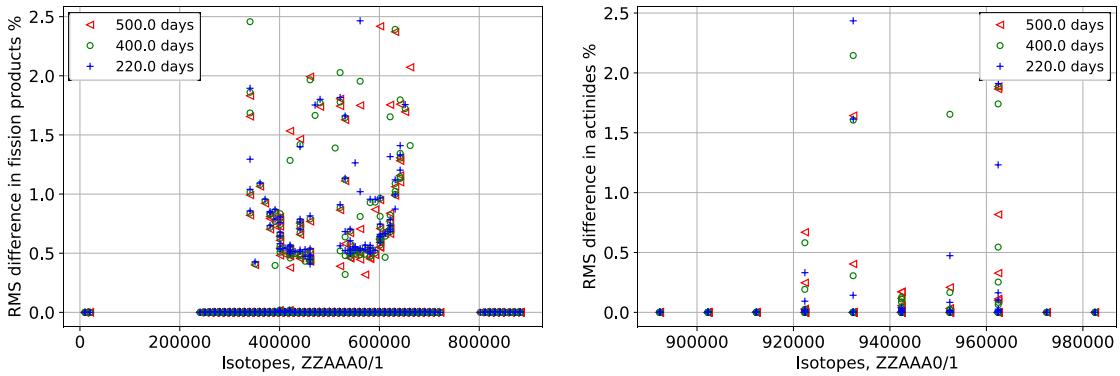
(d) Cm-244

Figure 6.31 Atomic Concentration and Difference Calculated by Serpent and DYN3D for Various Actinides

In order to understand the overall isotopic concentration difference, the RMS error for all actinides and fission products tracked are plotted in Figure 6.32 for three different points in the depletion cycle. In general, the results are in good agreement, with most of the isotopic difference below 1%. However, there are some fission products with differences in the 1-2% range and a small fraction of nuclides experience differences above 2%. The primary contributor to the depicted errors can be attributed to the mismatch in the spatial flux distribution between the codes. However, a fraction of the error in the concentrations of some fission products is due to the fixed fission yields used in this study.



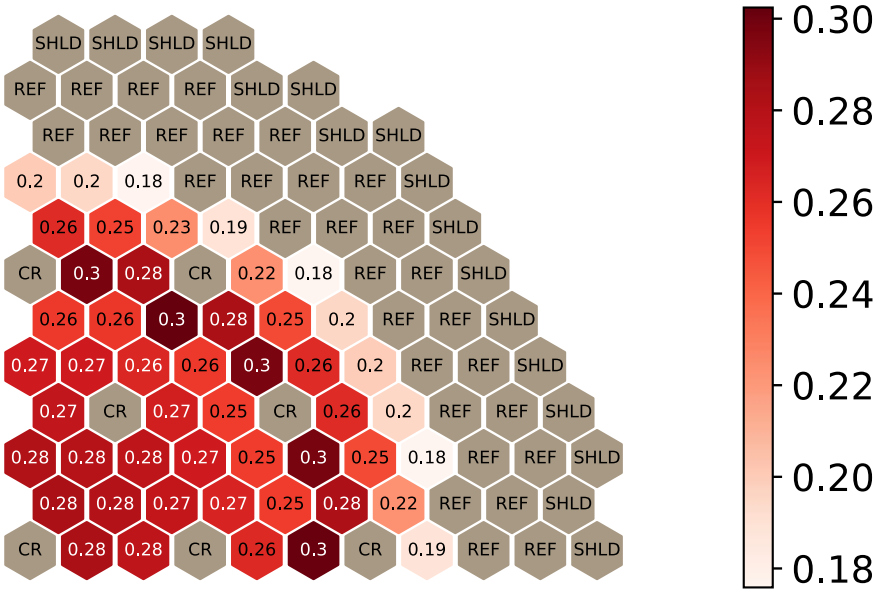
(a) Inner Core Region



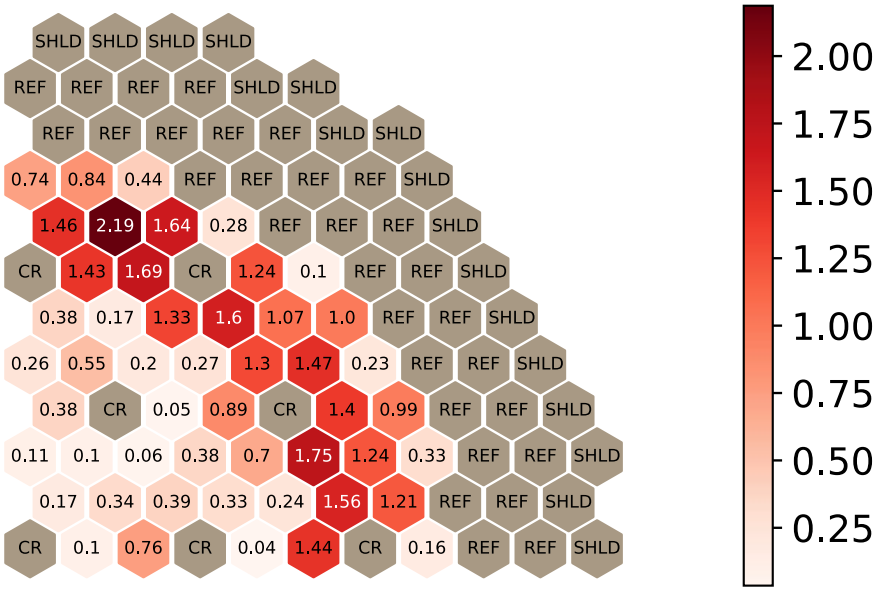
(b) Outer Core Region

Figure 6.32 RMS Difference in Atomic Density Calculated by Serpent and DYN3D for All Isotopes at Different Time-Points of The Cycle

Finally, the decay heat calculated by Serpent and DYN3D is compared in Figure 6.33 and Figure 6.34. First, the decay heat in each radial assembly channel is calculated. Results are presented for a time step at the middle of the cycle, when the relative power is at 90% of the nominal, and also at the end of cycle, where the relative power is at about 11% of the nominal.

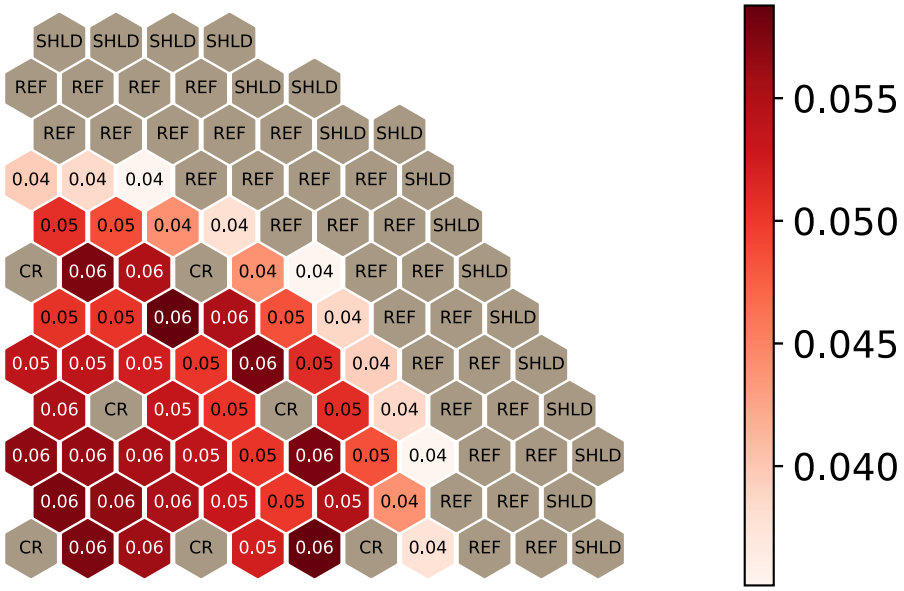


(a) Decay Heat Values (MW)

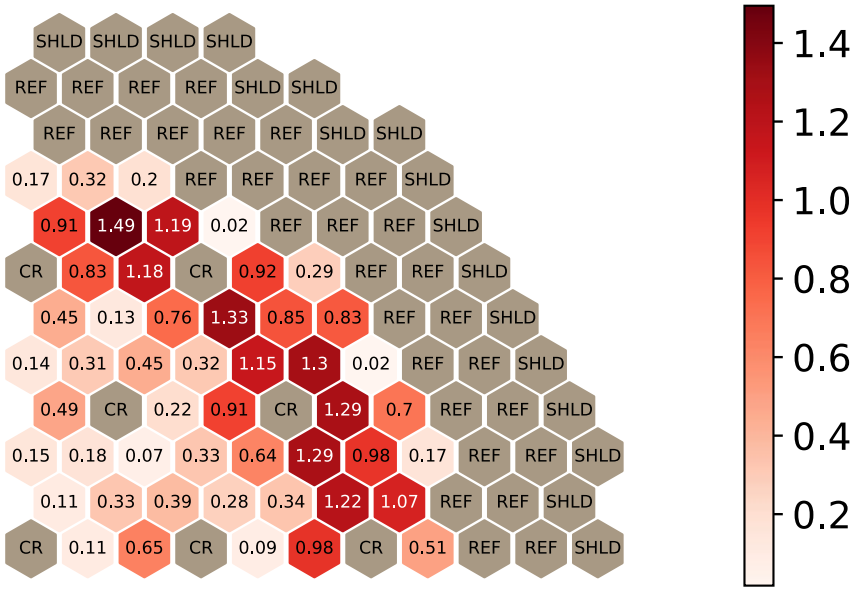


(b) Relative Difference in Decay Heat (%)

Figure 6.33 Decay Heat Calculated by Serpent and DYN3D and the Difference Between Codes in the Middle of the Cycle, 90% of the Nominal Power Level



(a) Decay Heat Values (MW)



(b) Relative Difference in Decay Heat (%)

Figure 6.34 Decay Heat Calculated by Serpent and DYN3D and the Difference Between Codes at the EoC, 11% of the Nominal Power Level

The radial comparison between codes demonstrates a very good agreement in both cases presented. A maximum difference in decay of around 2% was observed in the case in which the reactor power is at 90% of the nominal power, with most assemblies presenting a difference of around 1.5% and below. As for the case in which reactor power is low, close to 11% of the nominal, the observed difference in decay heat between Serpent and DYN3D is even smaller, with a maximum difference of 1.49%. It can be noted that, although great agreement is achieved for both cases, the difference in decay heat does tend to increase in the outer core region as compared to the inner core. This can be attributed to the higher fuel-reflector heterogeneity.

Lastly, the overall decay heat for the system calculated by each code is also compared and presented in Figure 6.35. As expected based on the previous agreement achieved, the overall decay heat calculated by Serpent and DYN3D is in a very good agreement, with a maximum difference of only 0.6%. This analysis demonstrates that even for a full 3D core, with a varying power profile, the Serpent-DYN3D code sequence is able to accurately and efficiently perform decay heat calculations.

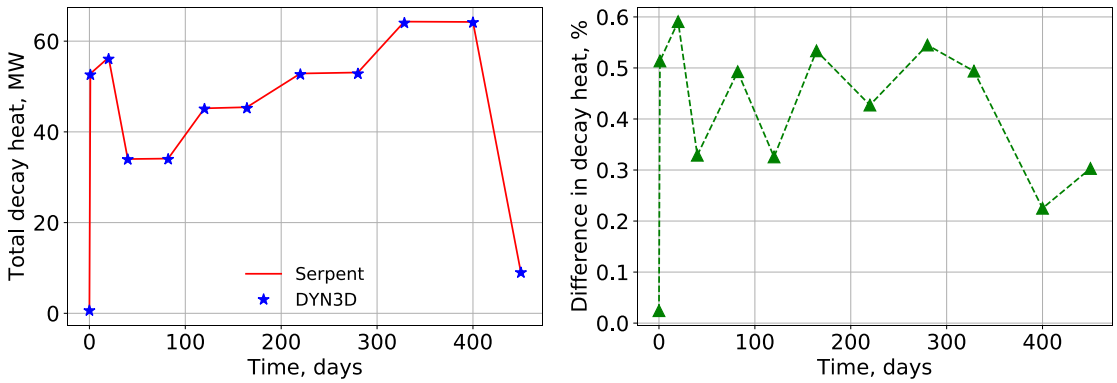


Figure 6.35 Total Decay Heat in the Core Throughout Cycle and Difference Between Serpent and DYN3D

6.5 DYN3D Performance Considerations

For this study, the core was divided into 900 burnable regions (180 fuel assemblies, each with 5 axial layers) both in DYN3D and Serpent. The Serpent reference solution was obtained using 500,000 neutron histories, with 200 inactive and 700 active cycles per depletion step. As a result, the maximum statistical uncertainty in power was on the order of 0.1%. It took 91 hours to complete the MC burnup calculations on 8 CPUs using the Open Multi-Processing (OpenMP).

DYN3D is a serial code typically executed on a single CPU. To improve the efficiency of DYN3D for fuel cycle analysis, a parallel version was created. The parallelization was implemented only for the depletion routine using the OpenMP. In order to perform a fair comparison, the same system and number of processors were utilized by DYN3D. The overall computational time of DYN3D was 28 minutes, in which the multi-group diffusion solution occupied 30% and the depletion calculations share was 70%. The overall time to generate the macro and microscopic cross-sections was ~30 hours. Although, the generation time seems to add significant overheads to the calculation time, the end use of the capability developed and demonstrated here must be remembered. Nodal diffusion codes are truly the only practical means to analyze a system under off-nominal scenarios. Such calculations mandate the use of tightly coupled neutronic-T/H calculations due to the strong variation of operational conditions, which affect the cross-sections, during the transient. The time to generate cross-sections then becomes negligible if compared to equivalent coupled calculations performed via the use of coupled MC-T/H sequences, which is prohibitively expensive and not practical. A major contribution is made in the current study by introducing a method to evaluate the decay heat using first principles. The

method allows to improve the fidelity of nodal diffusion codes to predict time-dependent spatial decay heat distribution, which is extremely important to analyze the system under transients. In fact, nodal diffusion codes are proven for their maturity and efficiency and thus are used as core simulators. Therefore, a true comparison should be made between the generation time to produce macroscopic cross-sections to that of generating both macro and microscopic quantities. Serpent was used to generate these cross-sections and a slowdown of ~10% was observed here.

In summary, it is found that performing the decay heat analysis using DYN3D leads to results that are in good agreement with the reference Serpent solution, while requiring significantly less computational resources.

CHAPTER 7. CONCLUSIONS

A summary of the work performed and the conclusions drawn are presented herein. Also, some possible aspects of work to be further developed are introduced.

7.1 Conclusions

The goal of this thesis was to establish a method of coupling Monte Carlo and nodal diffusion computational methods for the efficient analysis of sodium cooled, metal fuel fast reactors. The codes chosen for this work were the Monte Carlo code Serpent, and the nodal diffusion code DYN3D. The Serpent code was used for two purposes: to generate reference solutions for the cases studied, and as a cross section generating tool. Meanwhile, DYN3D was utilized for core simulations. Chapter 1 and Chapter 2 of this thesis provide an overview of the background and the theory applied in this work, including a brief description of neutron transport and neutron diffusion, as well as a description of the computational methods and the associated codes used in the subsequent analyses.

The core taken as a reference for this work is the Advanced Burner Reactor (ABR). In Chapter 3, detailed description of the reactor is presented, including details about each assembly type and materials used throughout the core. The full 3D ABR core is then modeled in Serpent to be used as a reference for comparison in further studies, as presented in Chapter 4. In order to verify that the Serpent model can be used as a valid reference, some static state calculations were performed and compared to results presented in the OECD/NEA benchmark. The results of the Serpent ABR model were within good

agreement to the average results by benchmark participants, verifying that the model could be used as a valid reference solution.

In Chapter 5, an initial depletion analysis comparing Serpent and DYN3D is performed for a 2D fuel lattice and for the full ABR core. Also, the process used for macroscopic cross section generation is presented. The analysis shows that for a 2D assembly, using the *INF* cross-sections leads to almost perfect agreement in criticality between Serpent and DYN3D. In the case of the full core, where the fuel assemblies experience closer-to-critical conditions, it is shown that the *B1* cross-sections generate better results, as demonstrated by comparing the criticality and the radial power distribution throughout the depletion cycle.

Finally, Chapter 6 presents the bulk of the work in this thesis, where the Serpent-DYN3D code sequence is applied for decay heat analysis. First, a study on the effects of operational conditions on the decay heat in fast reactors is performed. It is found that decay heat is not highly dependent on variations of fuel temperature and coolant densities, therefore the remaining of the analyses are performed using only the nominal operating conditions. The chapter goes on to describe the hybrid micro-depletion method implemented in DYN3D and the microscopic cross-section generation procedure. In Section 6.4, decay heat analyses were performed for a 2D infinite lattice, a 3D fuel assembly, and finally for a 3D full core model. In all the analyzed cases, very good agreement in criticalities, power distributions, isotopic concentrations, and spatial (and total) decay heat distributions were observed. The maximum difference in radial heat distribution was found to be 2.19%.

These results demonstrate that the Serpent-DYN3D code sequence is a useful tool in decay heat analysis, especially for advanced systems in which experimental data and correlations are not readily available. Applying this methodology saves time and computational resources, allowing for the analyses of various accident scenarios to be performed efficiently and with high fidelity.

7.2 Future Work

Some aspects might be taken into consideration for the continuation of this work. First, although the analysis has shown good agreement between the Serpent-DYN3D code sequence and the reference Serpent solutions, some steps can be taken to explore the possibility of diminishing the discrepancies even further. This can include the usage of the Superhomogenization (SPH) method [23], which is a process where the cross-sections generated in Serpent are corrected by applying an iterative method to take into account the difference between the heterogeneous and homogeneous fluxes. These differences might cause a larger impact in nodes where there is a large variation between the heterogeneous and homogeneous fluxes, such as in the case of radial reflective assemblies that are next to fuel assemblies. Another possible approach to diminish the difference between results might be related to the simplification regarding the fission yield as reported in Chapter 6. Generating fission yields that were energy-independent and constant throughout the fuel cycle led to non-negligible discrepancies of fission product concentrations. Therefore, future work should focus on developing approaches which avoid storing the fission yields as functions of operating conditions, which makes the associated data libraries prohibitive large.

Aside from further fine tuning the results presented in this thesis, the next logical step would be to apply the methods presented here to analysis of accident scenarios. The decay heat generated in an accident scenario is of upmost concern when evaluating the integrity of the reactor, and therefore the DYN3D transient capabilities can be explored for the analysis of accident scenarios of sodium cooled, metal fuel fast reactors. The ability to calculate the heat generated in each specific node allows for the determination of safety margins in relation to fuel melting in each subsection of the reactor. Another possible application of the Serpent-DYN3D code sequence for decay heat is to use it for the analysis of spent fuel management, either for storage or transportation of the spent fuel.

REFERENCES

[1] OECD Nuclear Energy Agency, "Technology Roadmap Update for Generation IV Nuclear Energy Systems," *Gen IV Int. Forum*, pp. 1–66, 2014.

[2] NEA, "Benchmark for Neutronic Analysis of Sodium-cooled Fast Reactor Cores with Various Fuel Types and Core Sizes, NEA/NSC/R(2015)9," no. February, p. 85, 2016.

[3] J. Duderstadt and L. Hamilton, *Nuclear Reactor Analysis*. Wiley, 1976.

[4] J. R. Lamarsh and A. J. Baratta, *Introduction to Nuclear Engineering*, vol. 23, no. 1. 1955.

[5] U. Rohde *et al.*, "The reactor dynamics code DYN3D - Models, validation and applications," *Prog. Nucl. Energy*, vol. 89, pp. 170–190, 2016.

[6] S. Duerigen, U. Grundmann, S. Mittag, B. Merk, E. Fridman, and S. Kliem, "A Trigonal Nodal Solution Approach To the Multi-Group Simplified P 3 Equations in the Reactor Code Dyn3D," in *M&C 2011*, 2011.

[7] E. Fridman and E. Shwageraus, "Modeling of SFR cores with Serpent-DYN3D codes sequence," *Ann. Nucl. Energy*, vol. 53, pp. 354–363, 2013.

[8] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, "The Serpent Monte Carlo code: Status, development and applications in 2013," *Ann. Nucl. Energy*, vol. 82, pp. 142–150, 2015.

[9] E. Fridman and J. Leppänen, "On the use of the Serpent Monte Carlo code for few-group cross section generation," *Ann. Nucl. Energy*, vol. 38, no. 6, pp. 1399–1405, 2011.

[10] J. Leppänen, "On the use of delta-tracking and the collision flux estimator in the Serpent 2 Monte Carlo particle transport code," *Ann. Nucl. Energy*, vol. 105, pp. 161–167, 2017.

[11] A. Johnson, D. Kotlyar, G. Ridley, S. Terlizzi, and P. Romano, "CORE-GATECH-GROUP/serpent-tools: 0.5.2a1 - Pre-release," Jun. 2018.

[12] T. K. Kim, W. S. Yang, C. Grandy, and R. N. Hill, "Core design studies for a 1000 MW th Advanced Burner Reactor," 2009.

[13] "Serpent - A Continuous Energy Monte Carlo Reactor Physics Burnup Calculation Code." [Online]. Available: http://montecarlo.vtt.fi/index.htm.

[14] D. A. Brown *et al.*, "ENDF/B-VIII.0: The 8thMajor Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data," *Nucl. Data Sheets*, vol. 148, pp. 1–142, 2018.

[15] E. Nikitin, E. Fridman, and K. Mikityuk, "Solution of the OECD/NEA neutronic SFR benchmark with Serpent-DYN3D and Serpent-PARCS code systems," *Ann. Nucl. Energy*, vol. 75, pp. 492–497, 2014.

[16] “ANSI/ANS-5.1-2005: Decay Heat Power in Light Water Reactors,” 2005.

[17] E. Shwageraus and P. Hejzlar, “Decay heat in fast reactors with transuranic fuels,” *Nucl. Eng. Des.*, vol. 239, no. 12, pp. 2646–2653, 2009.

[18] A. A. Gonchar and E. A. Rakhmanov, “Equilibrium distributions and degree of rational approximation of analytic functions,” *Math. USSR-Sbornik*, vol. 62, no. 2, p. 305, 1989.

[19] M. Pusa, “Rational Approximations to the Matrix Exponential in Burnup Calculations,” *Nucl. Sci. Eng.*, vol. 169, no. 2, pp. 155–167, 2011.

[20] M. Pusa and J. Leppänen, “Computing the Matrix Exponential in Burnup Calculations,” *Nucl. Sci. Eng.*, vol. 164, no. 2, pp. 140–150, 2010.

[21] Y. Bilodid, D. Kotlyar, E. Shwageraus, E. Fridman, and S. Kliem, “Hybrid microscopic depletion model in nodal code DYN3D,” *Ann. Nucl. Energy*, vol. 92, pp. 397–406, 2016.

[22] Y. Bilodid, E. Fridman, D. Kotlyar, and E. Shwageraus, “Explicit decay heat calculation in the nodal diffusion code DYN3D,” *Ann. Nucl. Energy*, vol. 121, pp. 374–381, Nov. 2018.

[23] E. Nikitin, E. Fridman, and K. Mikityuk, “On the use of the SPH method in nodal diffusion analyses of SFR cores,” *Ann. Nucl. Energy*, vol. 85, no. June, pp. 544–551, 2015.